Discovery of Morphemes in Swedish

Florian Eisl

Lund Institute of Technology
Department of Computer Science
Box 118
S-221 00 Lund, Sweden
x02fe@efd.lth.se

Abstract

This document describes an overview of two methods which allows to discover syntactic structures form a untagged corpora.

The first part describes the algorithm by Hervé Déjean – how it works, and shows the result of the algorithm applied to a Swedish corpus. A text by Selma Lagerlöf and a Swedish dictionary.

The second part shows the basics of the algorithms developed by Patrick Schone and Daniel Jurafsky.

1 Introduction

The morphological analysis is basically the segmentation of words into components that form the word by concatenation. From a practical point of view, the development of a fully automated morphology generator would be of considerable interest, since we still need good morphologies of many European languages and to produce a morphology of a given language by hand can take weeks or months. With the fact that a lot text is available online it is of great interest to develop morphologies of particular stages of a language, and the process of automatic morphology writing can simplify this stage, where there are no native speakers available.[1]

2 Hervé Déjean – Morphemes

The idea of Hervé is based on the approach by Harris and is characterized by two facts: (a) the use of corpora and (b) the use of the notion of distribution instead of the sense of elements. The distribution of an element is the set of environments in which the element occurs.

Only untagged and non artificial corpora without specific knowledge about the studied language is used. They try to discover the structures of a natural language from raw texts of this language. This kind of discovery is possible if there are some expectations of the structure of the Natural Language and some formal properties are used.

The method relies on structural linguistic concepts: the morpheme, the chunk and the linearity of the language, i.e. the corpus is

[1] Goldsmith, John: Unsupervised Learning of the Morphology of a Natural Language

composed of a unidimensional sequence of elements.[2]

2.1 Morpheme Discovery – How it works

The algorithm is based on the number of different letters which follow a given sequence of letters. The increase of this number indicates a morpheme boundary. For instance, after the English sequence direc, we only find, in our corpus, one letter t. After direct, we find four letters: i, l, o, and e (directly, director, directed, direction). This increase indicates a boundary between the root (direct and the suffixes (-ion, -ly, -or and -ed). The algorithm works well when the corpus contains enough occurrences of a stem family. But, it may generate wrong segmentations. For example from the list started, startled, startling, the algorithm outputs this segmentation: start-ed, start-led, start-ling. The errors occur when two kinds of stem families are used for the segmentation.[3]

The new idea for improving the segmentation now is to divide this operation into three steps. The first step computes the list of the most frequent morphemes. The second step is to extend this list by using the discovered morphemes already generated. And the third and last step is the segmentation of the words using the before produced morphemes. The illustration is only done for the segmentation of the suffixes but to get the prefixes the same algorithm can be used just with the reverse letters of the words.

2.1.1 Discover the most frequent morphemes

The aim is to find beginnings or endings of words which have the following property: after a given sequence of letters, we count the number of different letters. If this number is higher than a threshold (e.g. half the letters of the alphabet), we got a so called morpheme boundary, expect in the case that we are in the sequence which corresponds to another, to a longer morpheme, a case which can be detected. This can be illustrate by simple example, before the sequence "on" we found 20 different letters therefore "on" may be the morpheme. But 154 of these words in the used corpus end with "ion" out of 293 which and end with "on". Now it can be seen that the longest sequence "ion" represents more then 50% of the words ended by "on" and due to this it can be considered that the morpheme is not "on". "on" is only a part of the morpheme "ion".

The most frequent morphemes of the English and German language can be seen in the following table:

English	German
-e	-en
-s	-e
-ed	-te
-ing	-ten
-al	-er
-ation	-es
-ly	-lich
-ic	-el
-ent	

Table 1: The most frequent morphemes of English and German[4]

[2] Déjean, Hervé: Morphemes as Necessary Concept for Sturctures Discovery from Untagged Corpora

[3] Déjean, Hervé: Morphemes as Necessary Concept for Sturctures Discovery from Untagged Corpora

[4] Déjean, Hervé: Morphemes as Necessary Concept for Sturctures Discovery from Untagged Corpora

The most frequent morphemes of the Swedish language:

Swedish (selma)	Swedish (dictionary)
-ar	-erna
-er	-ningarna
-en	-ade
	-ar
	-ligt
	-ligast
	-ningarnas
	-ernas
	-ades
	-nings
	-ens
	-ers
	-ets
	-ad
	-ning
	-en

Table 2: The most frequent morphemes of the Swedish language

The first column is the result of text written by Thelma Lagerlöf. This corpus consists out of about 1.000.000 words.

The result in the second column is received using a Swedish dictionary as the corpus. The dictionary consists of about 120.000 words.

2.1.2 Discover other morphemes

After the most frequent morphemes of a language are found this morphemes can be used to find out other morphemes. This can be done using the following rule: For a given sequence of letters it can be checked if the next sequences of letters correspond to morphemes already found. If half of them belongs to the morphemes found, then the others can also be considered as morphemes of the language. This can be seen in the following table for the English language:

Morphemes found	words	New Morphems
	light	
-s	lights	
-ed	lighted	
-ing	lighting	
-ly	lightly	
-er	lighter	
	lightness	-ness
	lightest	-est
	lighten	-en

Table 3: Table of other morphemes of the English[5]

This algorithm is not perfect and also wrong morphemes are generated, but their frequency is very low. To make sure that we get only correct morphemes we use a threshold (five in practice). The morphemes with a frequency lower than the threshold are not found. The list of the received morphemes may greatly depend on the type of corpus used. The number of morphemes depends on the morphology of the language. What can be found out is, that morphemes have a similar behavior as words, a small number of them possesses a high frequency and corresponds to the mayor occurrences of the corpus.

2.1.3 Segmentation of the words

After all morphemes are found we use this morphemes to segment all the words in the corpus. The segmentation is done be using the longest match algorithm. This means that we segment each word with the longest morpheme that matches the beginning or ending of the word.

3 Patrick Schone and Daniel Jurafsky - Morphemes

A knowledge free algorithm which automatically induce the morphology structures of a language. The algorithm takes as input a

[5] Déjean, Hervé: Morphemes as Necessary Concept for Sturctures Discovery from Untagged Corpora

large corpus and produces as output a set of conflation sets indicating the various inflected and derived forms for each word in the language. An example for this can be the word "abuse". The result would contain the following words: "abuse", "abused", "abuses", "abusive", "abusively" and so on. The algorithm extends earlier approaches to morphology induction by combining various induced information sources: the semantic relatedness of the affixed forms using a Latent Semantic Analysis approach to corpus-based semantics, affix frequency, syntactic context and transitive closure. The algorithm achieves an F-score of 88.1% on the task of identifying conflation sets in English. The algorithm is also applied to German and Dutch and evaluated on its ability to find prefixes, suffixes and circumfixes in these languages.[6]

3.1 Morpheme Discovery – How it works

In the picture below an overview over this approach is shown.

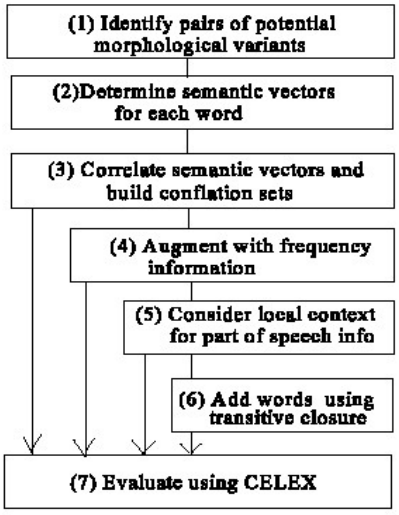

Figure 1: Overview how the algorithm works[7]

3.1.1 Identify pairs of potential morphological variants

The first goal is to find word endings which could serve as suffixes. A useful tool to find these suffixes is the so called character tree. Yet using this approach, there may be circumfixes whose endings will be overlooked in the search for suffixes unless we first remove all candidate prefixes. Therefore a lexicon of all the words in the corpus is built and all word beginnings are identified with frequencies in excess of some threshold (T1), so called pseudo-prefixes. All the pseudo-prefixes are stripped and the word residuals are added back to the lexicon. To show how the search for the suffixes works consider the following example. The following words are contained in the lexicon: align, real, aligns, realign, realigned, react, reacts, and reacted. Due to the high frequency occurrence of "re-" it is supposed to be a pseudo-prefix. If all the words are stripped of the "re-" and the residuals are added to a character tree the branch of the tree of words beginning with "a" can be seen in figure 2.

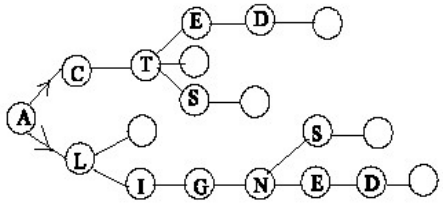

Figure 2: Character tree[8]

Out of the generated character trees rules can be received, but not all of these rules are correct and in the next step, incorporating semantics can help to determine the validity of each rule.

[6] Schone, Patrik and Jurafsky, Daniel: Knowledge-Free Induction of Inflectional Morphologies

[7] Schone, Patrik and Jurafsky, Daniel: Knowledge-Free Induction of Inflectional Morphologies

[8] Schone, Patrik and Jurafsky, Daniel: Knowledge-Free Induction of Inflectional Morphologies

3.1.2 Dertermine semantic vectors for each word

In order to obtain semantic representations of each word a singular value decomposition SVD is performed to a N*2N term-term matrix. The N represents the N-1 most-frequent words as well as a glob position to account for all other words not in the top N-1. The matrix is structured such that for a given word w's row, the first N columns denote words that precede w by up to 50 words, and the second N columns represent those words that follow by up to 50 words. Then the SVD is computed and the top 300 singular values to form semantic vectors for each word are kept.[9]

3.1.3 Correlate semantic vectors and build conflation sets

To make a correlation between these semantic vectors normalized cosine scores NCS are used. Out of these scores it is possible to get the probability that an NCS is random or not and it is possible to estimate the distribution of true correlations and number of terms in that distribution. These numbers are needed in the following step.

3.1.4 Augment with frequency information

If just a purely semantic-based approach is used the tendency is to select only the relationships with contextually similar meanings. To overcome this weaknesses of the semantic-based morphology induction the analysis can be improved by supplementing semantic probabilities with orthographic-based probabilities.

The motivation is now to use an approach based on minimum edit distance MED. Minimum edit distance determines the minimum-weighted set of insertions, substitutions and deletions required to transform one word into another. For example, only a single deletion is needed to transform rates into rate whereas two substitutions and an insertion are required to transform it into rating.

If this method for achieving the task is used the number of correct pairs of potential morphological variants PPMV can be increased by 3% than semantics alone had provided for the –s rule.[10]

3.1.5 Consider local context for part of speech info

There is no guarantee that two words which are morphological variants need to share similar semantic properties. Due to this it is possible to improve the performance if the induction process took advantage of local, syntactic contexts around words in addition to the more global, large-window contexts used in semantic processing.

There is an added benefit from following this approach. It can be also be used to find rules that though different, seem to convey similar information. This could be clearly be of use for part-of-speech induction.[11]

3.1.6 Add words using transitive closure

The algorithm contains semantic, orthographic and syntactic components but there are still valid pairs of potential morphological variants which may seem unrelated due to the corpus choice or weak distributional properties. In Figure 3 this property is demonstrated in greater detail.

By semantics only eight connections can be found starting at Abuse, abuse, abusers, abusing, …

[9] Schone, Patrik and Jurafsky, Daniel: Knowledge-Free Induction of Inflectional Morphologies

[10] Schone, Patrik and Jurafsky, Daniel: Knowledge-Free Induction of Inflectional Morphologies

[11] Schone, Patrik and Jurafsky, Daniel: Knowledge-Free Induction of Inflectional Morphologies

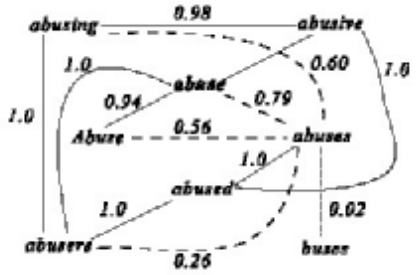

Figure 3: Semantic strengths[12]

3.1.7 Evaluate using CELEX

The algorithms are only applied to the words out of the corpus which have a frequency higher then 10. This cutoff slightly limits the generality of the results but it also greatly decreases processing time for all of the algorithms tested against it.[13]

4 Conclusion

Using Déjean's algorithm it is very important of which type the corpus is. As it can be seen in Table 2 two completely different results are archived using tow different corpora.

The corpus should be balanced and the result is becoming better the bigger the corpus is.

[12] Schone, Patrik and Jurafsky, Daniel: Knowledge-Free Induction of Inflectional Morphologies

[13] Schone, Patrik and Jurafsky, Daniel: Knowledge-Free Induction of Inflectional Morphologies

References

John Goldsmith. 2001. *Unsupervised Learning of the Morphology of a Natural Language*, University of Chicago, Association for Computational Linguistics.

Hervé Déjean, 1998. *Morphemes as Necessary Concept for Structures Discovery from Untagged Corpora*. Université de Caen, Basse Normandie,

Partick Schone and Daniel Jurafsky. 2001. *Knowledge-Free Induction of Inflectional Morphologies*, University of Colorado at Boulder.