From Semi-Automatic to Automatic Affix Extraction in Middle English Corpora: Building a Sustainable Database for Analyzing Derivational Morphology over Time

Hagen Peukert
University of Hamburg
hagen.peukert@uni-hamburg.de

Abstract

The annotation of large corpora is usually restricted to syntactic structure and word class. Pure lexical information and information on the structure of words are stored in specialized dictionaries (Baayen et al., 1995). Both data structures – dictionary and text corpus – can be matched to get e.g. a distribution of certain (restricted) lexical information from a text. This procedure works fine for synchronic corpora. What is missing, however, is either a special mark-up in texts linking each of the items to a certain time or a diachronic lexical database that allows for the matching of the items over time. In what follows, we take the latter approach and present a tool set (MoreXtractor, Morphilizer, MorQuery), a database (Morphilo-DB) and the architecture of a platform (Morphorm) for a sustainable use of diachronic linguistic data for Middle English, Early Modern English and Modern English.

1 Introduction

The sustainability of linguistic resources has gained considerable attention in the last years or so (Dipper et al., 2006; Rehm et al., 2010; Schmidt et al., 2006; Stührenberg et al., 2008). This development was probably initiated and was certainly fostered by federally funded research projects on information structure (SFB 632), linguistic data structures (SFB 441), or multilingualism (SFB 538). Work on sustainable language resources culminated in a row of frameworks, data models and structures, formats and tools. Also, it continues to prosper in related work (e.g. CLARIN). SPLICR, for example, addresses the issue of normalization of XML-annotated language records, or meta data (Rehm et al., 2010). In fact, the authors discuss the issues of a steadily growing proprietary tag set, the availability, the accessibility and the findability of linguistic resources. More precisely, search engines locate commercially produced data collections, but miss deep structured ressources of small research projects. Privacy and property rights restrict accessibility. Proprietary tag sets not obeying to established standards pose a problem for automatic analysis. In this vein, PAULA concentrates on stand-off annotations and the TUSEELDA repository states an example of integrated annotations. Both frameworks specify methods for handling and storing linguistic data. Finally, EXMERALDA is a tool for annotating spoken data in the first place. In sum, the focus in this field comprise work on annotation, format, tools, data integration (Dipper et al., 2006; Witt et al., 2009) and documentation (Simons and Bird, 2008). In a more general sense, these dimensions reflect Simons’ and Bird’s (2008) first three key players in a sustainable framework of language resources: creators, archives, aggregators, and users.

Although some of the repositories include historical language resources, the data structures and tools do not take into account the diachronic dimension, that is, language change over large spans of time is not represented in any of the models. Indeed, one finds tools for tagging morphological information, annotation schemas or tran-
scription (Dipper, 2011; Dipper, 2010; Dipper and Schnurrenberger, 2009), but they are not inte-
grated in the very architecture of the present frameworks. We like to initiate a kick-start to
close this gap by providing a first sketch of a plat-
form, a tool set and a database that is specifically
designed for diachronic data, i.e. adding the time
dimension. We will not elaborate on the issues of
annotations and formats here. For reasons of ease,
the annotation is kept as simple and as minimal as
possible so that they can be transferred to an ap-
propriate XML tag set, if available or necessary.

The Morphilo tool set aims at building a rep-
resentative diachronic database of English. The
software consists of three components: MoreX-
tractor, Morphilizer and MorQuery. MoreXtrac-
tor uses a quite simple algorithm that – dependent
on the given word class and a rule set – identifies
the structure of the word and assigns lexical tags
to it (e.g. /root or /pref). The identification pro-
cess is based on enumerated lists comprising all
prefix and suffix allomorphs listed in the OED.
After inputting a tagged corpus from a specific
time, MoreXtractor produces a text file, in which
the structure of all words is annotated.

Since the algorithm “overgeneralizes”, the file
has to be checked for wrong annotations. This te-
dious task is carried out by the Morphilizer com-
ponent. It takes each of the text files and its time
specification as an input, displays the word struc-
ture in a template and allows the user to make ad-
justments in a comfortable way by click and drop.
Each word item, its structure and its token fre-
quency that were checked manually are written
into the Morphilo-Database. For the given time
frame of the text, each word type has to be pro-
cessed only once.

MorQuery provides a comfortable search of the
database. Each combination of morphemes, allo-
morphs, compounds, word types, time frames or
corpora can be chosen from drop-down menus. It
is also possible to make selections of the most fre-
cquent queries or directly type SQL commands to
the prompt.

Last, Morphorm is a platform incorporating the
tool set and the database. Morphorm will be
available to the linguistic community on a web-
site. All researchers are encouraged to query the
data, but also to contribute to the project by hav-

```java
public enum SuffixEnum {
    ship("ship"), skiepe("ship"), scipe("ship"), 
    scype("ship"), scip("ship"), sipe("ship"), 
    shippe("ship"), schyp("ship"), schepe("ship"), 
    chepe("ship"), chyppe("ship"), 
    chep("ship"), shyppe("ship"), shipp("ship") 
}
```

Figure 1: representation of ship-suffixing their own diachronic corpora read in and ana-
lyzed. Since the database will have a large stock
of entries by its inception, the workload for post-
processing using Morphilizer for each additional
new corpus will be evanescently little.

2 Morphilo Architecture: Toolset –
Database – Platform

2.1 Data Structures
Prefix morphemes and suffix morphemes are
stored in enumerated lists. Each entry in the
list represents one morpheme referring to differ-
ing numbers of allomorphs. These allomorphs
were extracted from the OED (3rd edition, on-
line version). The OED enlists 179 entries for
prefix morphemes and 390 entries for suffix mor-
phemes. The various forms of each suffix – e.g.
mentt, mente, ment especially present in Middle
English – are referenced in the data structure as
allomorphs. In some extreme cases, such as the
prefix over-, the OED lists over 100 written vari-
ants. Other entries, such as the trans-prefix, have
only one form listed (see figure 1).

There are some cases in which one form repre-
sents several morphemes, e.g. there are three en-
tries for the ant-suffix. Since these cases are either
due to assimilation, misinterpretation (peasan(t),
for example) or meaning shift – all of which occur
over time – these cases are captured on the time
scale in the database (see section 2.2). The enu-
merated lists represent exactly one form of each
affix (morpheme) and all its allographs. Even
though there are some cases, in which an affix
form corresponds to several meanings at a time
(e.g. out Booij 2010: pp 19), this is clearly not
the rule, most likely a transitional stadium and
subject of an ongoing debate whether the same
meaning is involved. In addition, we find slight semantic differences in the majority of derivational affixes depending on the environment they are attached to. To illustrate, throughout the history of English, *lordship* has incorporated several meanings ranging from “a percentage on sales of books” (mainly used as such in the 19th century) to “arbitrariness” (documented in the 17th century) and “government, province, district” (OED, 2012). From the given example, it is clear that both the root *lord* and the *ship*-suffix embrace different meanings. So affix polysemy is as much a matter of degree as are slight semantic differences provoked by the semantic content of the “carrier word”. In sum, the enumerated lists alone do not include all necessary information, but need reference to the time information stored in the database. Equal forms referring to different semantic contents are represented at different time periods.

2.2 Morphilo Database

The Morphilo database is a MySQL-database and plays a pivotal role in the design of the application (see figure 2). It holds data on the position and order of derivational and inflectional affixes per predefined time slice (here 70 years). Moreover, compounds are included. They possess information on the position of the head and its type (e.g. exocentric, dvandva).

The basic unit of analysis is the word. In the corresponding table each analyzed word is listed

![Figure 2: ER-Diagramm of the Morphilo database](image-url)
once per time period. Along with the information of the word form, its root and part of speech are also given. If a word occurs more than once per specified period, the occurrence is incremented. The table occurrences is linked to the table corpora, which encodes the time information along with the name of the corpus to be analyzed. Time is specified by a beginning date and an end date. These dates are checked before the information of a new corpus can be added.

The compounds (figure 2) link to the corpora-table as well. However, compounds consist of words and hence compounds can be derived from the words-table. In the compounds-table itself, the order of its components (words) is encoded. All words, on the other hand, can be analyzed in terms of their components also, that is, affixes. The order of the affixes can be gained from the respective “position-tables”. For inflectional affixes, no position is specified. We assume for English that inflections occur at the end of the word only once. The tables prefix and suffix define all allomorphs whereas prefixmorpheme and suffixmorpheme harbor their morphemic representations.

Thus, simple as well as complex words are represented in a structured format. Each entry (compound or word) has to be analyzed only once per time period. Inconsistencies may arise if two different structures of the same word show up. In this case, one has to agree on one interpretation for the given time period.

2.3 Morphilo Toolset

The Morphilo tool set consists of three components: MoreXtractor, Morphilizer, and MorQuery. MoreXtractor commands a reductionistic logic matching a set of affix strings to the given word input by using a simple rule set of the English Morphology. Since this algorithm is highly overgeneralizing, the Morphilizer assists in correcting the overgeneralizations and storing the correct entries in a database. Last, MorQuery is a tool to conveniently query the database for all common features encountered in English derivational morphology. In short, the Morphilo tools assist in filling and querying the database.

![Figure 3: sample extract from a morphilo-tagged file](image)

2.3.1 MoreXtractor

MoreXtractor is a morphological tagger. For the present implementation, the program reads in Penn-Treebank-tagged text corpora and stores them in a vector. The graphical user interface (GUI) offers the option of processing word classes (N, V, A, or Adv). The POS-information is there to allow the user to filter the word classes of interest. Its effect on avoiding affixal ambiguity for internal processing is insignificant.

The software will then run a simple stemmer for the inflectional system of Middle English. The stemmer follows the logic of a 2-subsequential finite state transducer (Mohri, 1997) that aligns the known inflectional endings to the word. The archaic inflectional prefix y- is omitted. Likewise, the remnants of the Old English stem-based morphology as well as exceptions (ox-oxen, mouse-mice, sheep-sheep) remain unconsidered. All inflections are marked with /infl without any further encodings of the English inflectional morphology.

In a second step, each derivational prefix and suffix of the corresponding enumerated lists dependent on the word class is mapped to the stemmed item. Whenever several affixes can be fully mapped (e.g. -ion versus -ation), the longer item is selected because the probability that the longer affix corresponds to its lexical counterpart is higher (Best, 2003). Prefixes are mapped from left to right; suffixes from right to left. The remnant of the string alignments is tagged as /root. Last, the updated vector is stored in a text file (see figure 3).

One can clearly see that the transducer overgeneralizes. To be precise, the last entry in figure 3 – proper – the inflectional suffix -er, which usually specifies the comparative in adjectives, as well as the prefix pro-, which is eligible for nouns and adjectives, are indeed marked as affixes although they belong to the root of the monomorphemic word proper. In fact, this behavior of the algorithm is intentional because first it prevents us...
from missing any potential candidates by a manual follow-up analysis and second the algorithm is applicable to other languages more easily for its generality.

2.3.2 Morphilizer

Morphilizer organizes the final analysis by hand. MoreXtractor’s automated tagging procedure outputs a morphemically tagged output file. It is these annotations that will help the user to efficiently correct false affix annotations by click and drop and thus quickly build up a data stock that is then also used in subsequent matching procedures. Morphilizer’s design is based on the observer pattern (see figure 4). The affix interface is implemented by the Prefix, Suffix and Inflection classes, which register at the Select-class. The difference to the standard observer pattern is that the registered classes cannot resign from their “Observable” class once they are declared for a certain time period, that is, defined affixes stay the way they are (for more specific information please see the documentation section on www.morphilo.uni-hamburg.de).

Morphilizer takes three input variables: the tagged file, the time range and the corpus name. The algorithm starts by checking against the time range in the corpora-table of the Morphilo database (see figure 2). Once the specified dates fall within an existent time range and the corpus name is not yet included, all entries of the tag file are matched to the word-table referring to both its word class (POS in table word) and its word form (word in table word). If present, the occurrences of the item in the tagged file are counted, then deleted and the table occurrences is updated by incrementing the respective number in the field Occurrence. All entries that are not available in the Morphilo database are left unchanged in the file. They will be processed in the same manner as those corpora that fall outside any represented time range. For the latter case, the table corpora is updated first by the new corpus information (time range and name). Eventually the manual analysis begins.

Morphilizer presents each entry that is to be analyzed manually in text fields such as “prefix 1”, “prefix 2”, “root”, “suffix 5”, etc. corresponding to the automated analysis done by MoreXtractor. At this point, the user will interfere and either confirm or correct and rearrange the suggestions. Most of the commands in Morphilizer are carried out in this manner. Compound words undergo a slightly different procedure. Some of the Penn-Treebank-tagged corpora do not indicate compounds. Whenever real compounds occur in the word section of the Morphilizer GUI, they can be shifted to the compound section by a mouse click. At the end of the analysis, all instances of the corresponding item are counted and deleted in the original file. Finally, the new word is written into the database and all relevant tables are up-
dated. Deleting the entry from the original file enables the user to interrupt her or his work and go on at a later point in time. As a summary, the main sequences of the algorithm (MoreXtractor and Morphilizer) is visualized in figure 5.

2.3.3 MorQuery

MorQuery is the third component in the tool set. It is an independent program to query the Morphilo database more easily. A web-based interface is also available. In essence, the user makes a selection of the features of interests (corpus, types/tokens, word class, morpheme/allomorph, affix position, prefixes/suffixes/compound/words, derivation/inflection). The software combines these choices to valid SQL commands, queries the database and returns the results as textual output. The results can be saved for further statistical analyses in a tab-delimited format. While for very specific information requests, SQL queries can also be entered directly, a selection of the most common queries can be chosen from a drop down menu.

2.4 Morphilo Platform: Morphorm

Morphorm is a platform attempting to contribute to a sustainable framework of reusability of diachronic linguistic data. The framework incorporates the Morphilo tool set and the Morphilo database. In addition, it extents the prevalent structure to meet the requirements of a multi-user design. The main idea behind Morphorm is similar to web wikis: share work - receive full profit. Users contribute to the data stock and profit themselves from a more representative set of data and less annotation work. With each additional unit of annotated text, future annotation work will be substantially less for all users since each item (word or compound) has to be analyzed only once.

Figure 6 depicts the architecture. Note that MoreXtractor receives direct input on the time range and words from the database here. This

Figure 5: Architecture of Moreextractor and Morphilizer
A feature is part of Morphilizer in the standalone application. Also, a list of analyzed corpora ensures that no data is processed twice. Each new corpus is written to this file. The second difference in Morphorm is that new data is not written into the original database, but to separate datasets that are structurally identical. The third adjustment made in Morphorm is quality control. Decentralizing quality control is a sensitive issue and cannot be fully automated. There is no full-fledged solution available, but we will use indicators and reported feedback by users. A first indicator is the frequency of usage of a certain dataset by the user community and in publications. A high frequency indicates a certain trust in the analyzed data. A second indicator is, if available, data of the registered user, e.g., his or her project, background, or department. Third, unexpected differences in the result sets of the Morphilo and the user dataset hint at possible erratic annotations. However, from the suspicious datasets a sample will be drawn and will be checked manually. Last, reported errors from other users will contribute to revising or excluding datasets from accommodating it to the master file. If a “user dataset” meets all quality standards, it is incorporated into the Morphilo database.

The integration of MorQuery made an additional selection field necessary. The user makes the choice on a selection of datasets most suitable to her or him. The quality of the Morphilo database is assured; for all others that have not been checked for quality no guarantees can be made. So, it is up to the user to make a decision on the trade-off between representativeness and risk of wrong annotations. A possible way of dealing with this situation is to make several queries (similar to the procedure described above for quality control): one with the Morphilo dataset, one with all datasets and one with the personal selection of datasets. If the results deviate substantially from the Morphilo results, the selection should be treated with caution. The data should be checked individually and reported to the quality control.
3 Discussion

A first criticism could be addressed to ignoring the XML standard for making morphological annotations and a respective XML-based repository. There are two lines of argumentation to support the present configuration. First, MoreXtractor produces output for Morphilizer. The output is not meant as a tagged text for further external processing. Really, the annotation is added for reasons of user convenience. It is indeed possible to use an XML schema instead, but it does not justify the effort because the database, at least not now, is not represented as an XML repository. This leads to the second line of argumentation, it is still unclear whether XML in its present implementation will be established as a standard for linguistic annotation in general. At present, the “Morphilo data” is available in a structured format. It is unproblematic to transfer MySQL data or object data to XML subsequently if agreement on a standard is reached. Until then, it has advantageous for programming and available design patterns to use the present structure.

In the light of the recent developments of word taggers, a second criticism could be directed towards the simplicity of the algorithm of MoreXtractor. Again, the idea behind MoreXtractor is not to give a reasonable text output for further external processing. More importantly, the software is not tailored for one particular language. Even if the present implementation is for the English Language, the Morphilo framework as such could be implemented in any other language, in which derivational morphology is an important part of the grammar. A simple matching procedure that depends on word class affixation as its only constraint can be implemented for any language. In contrast, from a typological perspective, the idiosyncrasies of language-specific morphology is the most complex. Hence an architecture heavily dependent on language-specific morphology results in a large effort of adjustments.

Finally, the success of the Morphilo crucially depends on the participation of other scientists in the field of the historical derivational morphology of English. Supposedly, the number of these scientists cannot be exceedingly large and so shared annotation work will only pay off over a larger time frame. In this case, success requires great persistence and obviously it implies data sustainability. In addition, a larger time horizon could pose an issue to quality assurance as well because it entails maintenance and as such man power. We can only speculate on the future acceptance of Morphilo, but once the initial database comprises the bulk of the known vocabulary of Middle English and Early Middle English, only very few new words will continue to be incorporated so that maintenance is then to be restricted to a minimum. At this stage, we will have arrived at a nearly fully automatic affix extraction device for derivations and inflections.

4 Summary

We have presented a tool set that helps to analyze lexical units and organize the work on historical text corpora. These tools can also be used in a web-based platform encouraging a culture of sharing and participation, but also saving time and work. The idea grew out of the need to cooperate more intensively in the field of historical linguistics on the basis of digital texts and media. From some publications in the field (Hiltunen, 1983; Dalton-Puffer, 1996; Haselow, 2011; Ciszek, 2008; Bauer, 2009; Nevalainen, 2008) and personal communication we can see that annotation work of the same corpus material is often carried out several times. In fact, often conflicting evidence is produced because of deviant procedures in the analysis of data.

By initiating a platform and making it known to the research community, not only the workload can indeed be diminished, but also a common standard for analyzing diachronic derivational affixes can be established. At the same time, large and more representative sets of diachronic linguistic data allows us to apply a larger spectrum of quantitative methods. As a consequence, the successful implementation and acceptance contributes without much ado to a sustainable use of historical linguistic data. It is in this spirit that we like to recommend the Morphilo framework to other scientists in the field.
References

Risto Hiltunen. 1983. The decline of the prefixes and the beginnings of the English phrasal verb: the evidence from some Old and Early Middle English texts, volume 160 of Turun yliopiston julkaisuja. Turun yliopisto, Turku.
Thomas Schmidt, Christian Chiarcore, Timm Lehmborg, Georg Rehm, Andreas Witt, and Erhard Hinrichs. 2006. Avoiding data graveyards: From heterogeneous data collected in multiple research projects to sustainable linguistic resources.