1 QAMPLD

1.1 About QAMPLD

In this chapter, we introduce our software *QAMPLD*[1]. *QAMPLD* originated from the idea of supporting our theoretical research with the retrieval of statistical data for quantitative analysis. *QAMPLD* [kwəmpld] is an acronym and stands for ***Q**uantitative **A**nalysis of **M**orphological **P**roductivity and **L**exical **D**iversity*. As you may expect from the name, the main task of the application is to support quantitative analysis of the productivity of morphological word formation processes and also the comparison of the lexical diversity of different texts from different authors.
We chose the programming language *Python*[2] with the *Natural Language Toolkit*, *NLTK*[3], for implementing *QAMPLD* as a sophisticated, standalone command line tool. Implementing *QAMPLD* in *Python* allows the usage on different operating systems and *NLTK* offers access to a wide range of powerful and highly optimized libraries for natural language processing. With more than one million words from 500 different texts, the *Brown Corpus*[4] is the fundamental source of text-data for *QAMPLD*. As it provides manual reviewed part-of-speech tagging information, it is a reliable source for conducting fine-grained statistical analyses. Still it is possible to use any other kind of corpus with *QAMPLD*, as long as it provides reliable part-of-speech tagging information. The role of correct part-of-speech tagging information in *QAMPLD* will be explained in sections 1.2.2 and 1.3.2.

1.2 Lexical Diversity

1.2.1 Field of Application and Usage

QAMPLD allows the comparison of any two individual texts from the 500 samples available in the *Brown Corpus*. For that purpose an enumeration of all 500 texts with basic information about the content length supplemented with an extract of the actual text is displayed for selection (see Screenshot 3). This enables a user to decide on which two texts are reasonable to compare in consideration of the content length. This is important, as it has already been highlighted in section 1. In fact, the lexical diversity of shorter texts is commonly higher than the one of longer texts. To facilitate this decision the list of available samples is ordered by content length.
After selecting two distinct samples, a user is presented with different types of information. The first part gives him the lexical diversity of the first text opposed to the one of the second text. As further information the usage of the most frequent words of each text is compared with their frequency in the other text. To neglect the usage of function words and for enhancing the significance of the presented data, the comparison is done individually for words of the four basic word classes: Nouns, verbs, adjectives and adverbs.

1.2.2 Functional Principle

The basis for determining the lexical diversity in *QAMPLD* is the type-token ratio as introduced in section 1. However it is necessary to preprocess the text-data in order to enhance the accuracy of automatic calculations. Especially when implementing the approach in software, a programmer may give into the temptation of building statistics purely on the occurrences of strings in a text. This, however, would not be sufficient. There are several reasons why.
One is the fact, that not every character or sequence of characters represents an actual word. Hence, punctuation characters and numerical characters are stripped from the text-data in *QAMPLD*.
Another reason why simple comparison of unprocessed strings can cause a distortion of statistical information is inflectional morphology in the English language. Thus an author may use the same word again but in an inflectional form. In English there are inflectional forms of nouns and verbs. Nouns can have a plural and a possessive form, while verbs can have distinct forms in different tenses. For instance, *eat* and *eaten* represent the same verb in different tenses, *child* and *children* are the

[1] Project website: http://qampld.tk/
[2] http://python.org/
[3] http://nltk.org/
[4] http://nltk.googlecode.com/svn/trunk/nltk_data/index.xml

singular and the plural of the same noun. Consequently, the repeated usage of the same word in a text should be detected regardless of its inflectional form. Therefore, grouping different inflected forms together is required.
The approach taken for that matter in *QAMPLD* is the utilization of the *WordNet lemmatizer*[5], which is included in the NLTK library. For proper functioning, the lemmatizer is fed with part-of-speech tagging information alongside the actual word. It is important that this part-of-speech tagging information is correct, since otherwise a wrong lemma of a word could be formed. For example the word *said* is lemmatized correctly as *say* when indicated as a verb, however when it is tagged as a noun, the lemmatizer would return *said*, therefore grouping of *say* and *said* wouldn't be possible anymore.

1.3 Morphological Productivity

1.3.1 Field of Application and Usage

QAMPLD's main field of application is the analysis of morphological productivity of word formation processes with certain suffixes in the English language. There are 50 predefined suffixes in *QAMPLD* from the word classes noun, verb, adjective and adverb. For measuring productivity, the growth rate as introduced in section 1 is being used in *QAMPLD*.
When launching the application a user can choose from two different options for analyzing morphological productivity (Screenshot 1). The first option he can choose is the *single suffix statistics*. This will then present him a list of all 50 suffixes available in *QAMPLD* (Screenshot 2). For each suffix he gets additional information about the word class in which it is being used. This is necessary because some suffixes like *-al* or *–ant* exist for different word classes, in this case for nouns and also for verbs.
After selecting the desired suffix, the result of the analysis is being displayed (Screenshot 6). The first value shown is the number of tokens in the corpus that have the corresponding suffix. This means, any occurrence of a word ending in the suffix is counted. The next information is the count of types with this suffix. For this figure each word is only counted on its first occurrence. As further information, the number of hapax legomena ending in the suffix is displayed. These are all the words that only occur once within the entire text. This figure together with the number of tokens is also used to calculate the next value, which is also the most important one - the growth rate. This value gives an estimation of the productivity of the suffix. As explained in section 1 the growth rate is the probability of the occurrence of a new hapax legomenon in a text. So, the higher this value, the more likely it is, that the suffix is productive. As last information, an overview of the most frequent words formed by the suffix is displayed as a frequency distribution list.
The second option that can be chosen in the menu is to compare the productivity of different suffixes. For that purpose the list with all available suffixes is shown. This time however, it is possible to select more than one suffix. One can select either several arbitrary suffixes by entering their corresponding number in the list or by selecting a specific word class. It is also possible to select all available suffixes at once.
After selection of the corresponding suffixes and a - on the amount of suffixes depending - processing time, one can choose to sort the results alphabetically, by growth rate, by token count, by number of types, or by the number of hapax legomena. Consequently a sorted list of all chosen suffixes is shown, displaying information about their individual word class, their number of tokens, types, hapaxes, as well as their calculated growth rates (Screenshot 5). If desired, one can also change the field by which the table is to be sorted at this point. This is especially useful when a large amount of suffixes has been chosen.

1.3.2 Functional principle

QAMPLD measures the productivity of word formation processes by calculating the growth rate for different suffixes of the English language. However, before this calculation can be done, it is important to identify the words that belong to the morphological category we want to examine, i.e. we want to find all members of the suffix in question.

---

[5] http://nltk.googlecode.com/svn/trunk/doc/api/nltk.stem.wordnet.WordNetLemmatizer-class.html

Hereafter, we will discuss the different approaches taken by *QAMPLD* to increase the accuracy of matching corresponding suffixes to their members. By way of illustration, we are going to give examples of changes in the amount of identified members after each method has been applied. In the following we are using the suffix *–ate*, which is used for forming verbs, as an example.

Initially:

When we just search the non-preprocessed corpus for words ending in *–ate* we get:

a) Use part-of-speech information:

Now, to refine the results, *QAMPLD* uses the part-of-speech information that is stored within the corpus to analyze only those words, whose word class matches the one of the word formation process of the suffix.

This decreases the amount of matches for *-ate* significantly:

The reason is that there are two other suffixes *–ate* which are used for forming nouns and adjectives.

b) Lemmatize words

In section 1.2.2 we already discussed that the lemmatization of words eases the detection of word-repetition. Beyond that, lemmatization also plays an important role with regard to the recognition of suffixes. Lemmatizing strips inflectional suffixes from a word so that derivational suffixes are exposed at the end of the word. This is especially important for verbs like these formed by the suffix *–ate*. In most cases they don't appear in their infinitive and therefore won't end with *–ate* but rather with e.g. *–ated* or *–ates*. This also explains the huge growth of identified tokens for *–ate* after lemmatizing:

c) Ignore words equal to suffix

QAMPLD ignores words that solely consist of a sequence of characters that are the same as the suffix itself. It is obvious that appending a suffix to an empty string is not a morphological word formation process. Words that are of the same sequence of characters as certain suffixes of the same word class as the word itself are: *Able, ant, hood, ship, ness, ling, let, ism* and *ion*. Since there is no infinitive version that is equal to *ate*, the token count of our example does not change in this case.

d) Ignore words one char longer than suffix

QAMPLD also ignores words that are exactly one char longer than the suffix that is being searched for. The reason is the same as with words that are equal to the suffix. There is simply no word formation process based on a single character.

Since verbs like *date, hate* or *rate* are now ignored there are changes in token and type count for our example *-ate*:

e) Ignore words with only vowel in suffix

QAMPLD also ignores all occurrences of words that end with the sequence of chars of the suffix but despite of this sequence have no other appearance of a vowel or a *y*. The reason is, that in English there are practical no words without at least one vowel or a *y*. And for the very few exceptions it is most unlikely that they are being used for morphological word formation processes. Under this assumption we can further enhance our algorithm to not detect those false positives. In the case of *–ate* there are exactly three words that are filtered out by this method. They are *state, skate,* and *grate*. The changes in the measures for *-ate* are then as follows:

Manual evaluation

At this point we discussed all current measures taken by *QAMPLD* to identify members of a suffix. Finally with these methods we achieved a reasonable rate for *-ate* but still with 20% false positives as a manual evaluation reveals:

As you can see from the figures, 77 false positives have been removed after manual correction of the data. This data was obtained by checking dictionary[6] entries for each word in order to find out the origin of the word. The process revealed, that 290 of the inspected words were really formed by the suffix *–ate*. From the 77 false positives 54 words had Latin origin and weren't retrieved by word formation processes. Examples are *collate* from *collatus* or *negate* from *negatus*. Another 2 words were from French origin: abate from *abatre* and debate from *debatre*. A total of 16 words were formed as a result of backformation. For example *manipulate* originated from *manipulation* and *donate* from *donation*. The last 5 words were compounds of other words as for example *understate* from *under* and *state*, or *overrate* from *over* and *rate*.
Most importantly however, is the fact that the growth rate hasn't changed significantly after manual correction. We can accept a growth rate of about $P \approx 0.03$ as a result of our analysis. This value suggests, that *–ate* is not as productive as for example the suffix *–ize* which has a measured growth rate of $P \approx 0.05$ but it still has a better growth rate than *–fy* which only has a value of $P \approx 0.02$.

1.4 Future Work

QAMPLD is open for future extensions and improvements. Especially the range of functions for measuring the lexical diversity could be increased. Currently it is only possible to compare two texts at the same time. Of course it would also be interesting to compare several texts at once. Statistics about all texts to see the difference between the text with the highest and the text with the lowest lexical diversity would also be interesting. This would however require adjustments of the content length for some texts to be accurate.
The morphological productivity functionality of *QAMPLD* could be enhanced in the manner of recognizing members of a suffix. Especially a reduction of false positives would be beneficial. As seen in the analysis of the *–ate* suffix there were still 20% false positives. These however didn't influence the growth rate significantly. One approach to accomplish a reduction could be the automation of some of the tasks that have been done in the manual correction of the data. We used information provided in an online dictionary to determine the origin of a word. This gave us the basic information whether the word was formed by a word formation process with the corresponding suffix or if it originated from a different language or from a backformation. Since this information was in a standardized format - HTML to be exact - we were able to automate some parts of our manual research. We loaded the webpages for each of the words in Python and parsed the information given by the dictionary to determine if the word was formed with this suffix. This then gave us the information about every word that was correctly recognized. The other words we then had to check manually.
If one would accept the accuracy of the information provided by the online dictionary, one can definitely improve the accuracy of the type count for some suffixes. The automation however would require an individual parsing of the web information for each suffix available in *QAMPLD*. Therefore for each suffix a Parser would need to be implemented and validated. Though, doing this for 49 suffixes requires a lot more effort than for just one suffix.
Another potential for improvement is the recognition of types that have been involved in more than one word formation process and which are then recognized by *QAMPLD* as two different types for the same word formation process. This happens when a prefix is added to a word that has been deduced from a word formation process that appends a suffix. Currently words like *activate* and *deactivate* are recognized as different types. Considering however, that *deactivate* has been created by prefixation of *activate*, it would be advantage to count both words only as one type when analyzing the productivity of *-ate*.

---

[6] We used http://dictionary.reference.com/

2 Conclusion

In this paper we approached the basic principles of quantitative measures for morphological productivity of word formation processes and of lexical diversity to form a theoretical framework for introducing our software *QAMPLD*. In a case study we showed that even with a rate of 20% false positives for the verb-forming suffix *-ate*, we still had a plausible accuracy for the calculation of the growth rate, compared to the result of a manual evaluation. These measures suggest for verb-forming processes that the suffix *-ize* is more productive than *-ate* but still *-ate* to be more productive than the suffix *-fy*. To share our application with the community, we published the source code of *QAMPLD* on the project website[1].