Report on Renaissance Musician Extraction from Baker's

INTRODUCTION:

The goal was to get a list of names and dates of Renaissance (1400 - 1600) musicians from Baker's Biographical Dictionary of Musicians (1901 edition). The book was obtained from archive.org, and ABBYY was used to perform OCR on it. ABBYY was selected because it is able to recognise bold characters, which is reflected in the output using HTML tags. Since all the names in the dictionary are in bold, ABBYY would make extracting names trivial. However, it was not as trivial as it was initially assumed, which is why another method had to be used.

METHODS:

It all begins the same way. When Baker's was given to ABBYY, its detection language was set to English. Italian, German, and French were also added so that most special characters could be recognised. After the HTML output was produced, two methods were used to get the names and dates.

METHOD 1:

Using an HTML parser, all text wrapped in tags specifying boldface font were extracted. The extracted text was then ran through a filter to name sure they were names.

PROBLEMS OF METHOD 1:

Many of the names in the HTML output were not in boldface. Often, only half a name (last name or first name) would be bold, and much of what was bolded was not supposed to be. Thus, many names were ignored since they were not bold, or were filtered out since they were incomplete. At this point, method 1 was abandoned for method 2.

METHOD 2:

Baker's is organised like a dictionary, and thus, each entry has its own section that resembles a paragraph. In ABBYY's HTML output, this is realised by wrapping each section in ... tags. Using an HTML parser, the text was broken up into sections according to these tags. For each section, a regular expression (included in Appendix A) was used to get the name in the article, and the name was run through a filter to make sure it was actually a name. Birth and death dates were also extracted using a regex. Finally, a list of Renaissance musicians were compiled using birth/death dates.

PROBLEMS OF METHOD 2:

* Not the beginning of every section contained a name because a section of text was often broken up at the bottom of a page to continue of the next. Thus, it was necessary to filter out the beginnings of sections that were not names. Unfortunately, this filtered out many legitimate names like Palestrina (since he only has one name); Beet'hoven [bät'hö-vn], Ludwig van (last name too long).

* The HTML parser used did not put spaces between bold text and normal text, resulting in "Hasler(orHassler)" where both "Hasler" and "Hassler" are bold, but "or" is not.

* The formatting of the dictionary is not consistent. In general, birth dates are written in the form “b. City, Country (occasional additional notes) Date”, though “b.” is sometimes “born”. “d.” and “died” are used for death dates, which are otherwise formatted the same way. When something is unknown, it is either replaced by “(?)” or simply left out. “(?)” is also used to indicate uncertainty. When looking for dates, the regular expression matched “b.” or “born” and “d.” or “died”, followed by a specific amount of characters, followed by a year. The number of characters between the beginning of the expression and the year is constrained to avoid matching sentences like “b. Antwerp. Eminent lutenist of the 16th century; pubi., 1592”. Unfortunately, this means that valid dates are sometimes not matched (“born at Zelazowa Wola [Fol. Jeliasovaya-Volia], a village near Warsaw, on Feb. 22, 1810”).

* ABBYY sometimes did not read numbers correctly, with “1” being read as “l”, “9” as “g”, etc. The regular expression was written to take mistakes like this into consideration without being too inclusive. Of course, this calls the reliability of the dates into question (though they should all be within an uncertainty of 100 years). The dates were checked to make sure that the birth date was less than the death date. Also, if the difference between the two was greater than 80, or less than 30, they were manually checked for correctness.

APPENDIX A

REGULAR EXPRESSIONS USED:

Regular expression to match names:

```
"^(?P<last>.{,%s}?), ?(?P<first>.{,%s}?)," % (maxlast, maxfirst)
```

where maxlast is the maximum characters allowed in the last name, and maxfirst is the maximum characters allowed in the first name (including middle names).

Regular expression to match birth year:

```
"[.,;\-\(\) ]([bh][\.,]|born ).{,%s}?[.,;\-\(\) ](?P<birth>[1lIi][0-
9lJIgiSo]{3})[.,;\-\(\) ]" %(maxchar,)
```

and death year:

```
"[.,;\-\(\) ](d[\.,]|died ).{,%s}?[.,;\-\(\) ](?P<death>[1lIi][0-
9lJIgiSo]{3})[.,;\-\(\) ]" %(maxchar,)
```

where the variable maxchar is the maximum number of characters allowed between “b.” (or others) and the corresponding year.

CALCULATION FOR PERCENT OF NAMES EXTRACTED:

Choosing 10 random pages in Baker's and counting the number of people on each page gives 87 people. That gives an estimated 8.7 people per page. Around 650 pages of Baker's have entries, giving a total of 5655 people in total. 5098 names were extracted, which is about 90% of the total amount.