Enhancing Access to the Levy Sheet Music Collection: Reconstructing Full-Text Lyrics from Syllables

Brian Wingenroth, Mark Patton, Tim DiLauro
Digital Knowledge Center
Johns Hopkins University
3400 N. Charles St.
Baltimore, MD 21218
{wingenroth, mpatton, timmo}@jhu.edu

ABSTRACT

The goal of the Lester S. Levy Sheet Music Collection, Phase Two project is to develop tools, processes, and systems that facilitate collection ingestion through automated processes that reduce, but not necessarily eliminate human intervention[1]. One of the major components of this project is an optical music recognition (OMR) system[2] that extracts musical information and lyric text from the page images that comprise each piece in a collection. It is often the case, as it is with the Levy Collection, that lyrics embedded in music notation are written in a syllabicated form so that each syllable lines up with the note or notes to which it corresponds. Searching the syllabicated form of words, however, would be counterintuitive and cumbersome for end-users. This paper describes the evolution of a tool that, using a simple algorithm, rebuilds complete words from lyric syllables and, in ambiguous cases, provides feedback to the collection builder. This system will be integrated into the workflow of the Levy Sheet Music Collection, but has broad applicability for any project ingesting musical scores with lyrics.

Categories and Subject Descriptors

H.3.1 [**Information Storage and Retrieval**]: Linguistic Processing

Keywords

Music information retrieval, text segmentation

1. INTRODUCTION

Sophisticated search capabilities improve a user's ability to take full advantage of large digital collections. Access to digitized musical scores can be enhanced significantly with the addition of searchable lyrics; however, manually entering lyrics is a time-consuming process. One less costly alternative is using optical music recognition (OMR)[2] to extract the lyrics from the sheet music. Unfortunately, lyrics appear in sheet music as a sequence of syllables and not as a sequence of words. This presents a problem because of the difference between how the text is stored and how users perform searches. One possible solution is to syllabicate – or break into syllables – each search query and match the syllables against the stored representation of the music. This approach would increase the overhead of every search. Instead, we reassemble the lyrics into words and store them – a process that is performed only once for each piece. This process of reconstructing words from syllables is a variant of the text segmentation problem. Previous work in text segmentation has focused on Asian languages that have no word boundaries such as Chinese and Korean. Reconstructing words from syllables is a similar problem. The goal is to eliminate the syllable boundaries that are not also word boundaries. We have explored two methods of reassembling words from syllables.

2. SYLLABICATED DICTIONARY

The basic operation supporting the syllabicated dictionary system is the retrieval of a set of words containing a given syllable. This operation requires a comprehensive list of syllabicated words. Only languages for which such a list exists can be used with this system. We use *Webster's 1913 Dictionary*[4] as a source of English words and their syllabications.

Given a syllable of our input, we retrieve the set of words containing that syllable, the set of words containing the previous syllable, and the set of words containing the next syllable. We then join the word set of the previous syllable to that of the current syllable, and join the word set of the current syllable to that of the next syllable. The union of these two sets is the possibility set for the current syllable.

From these possibility sets we find each path, where a path is a sequence of words that segments the input. There may be several paths since consecutive syllables can form different sets of words depending on the word boundaries chosen. For example, consider: *gen tle man*. These syllables can form one word, *gentleman*, or two words, *gentle* and *man*.

The system's initial performance was poor because test data contained many words not found in our dictionary. The majority of those unknown words were either plurals, verb conjugations, proper nouns, or words with different syllabications than Webster. Unknown words are a limiting factor

JCDL'02, July 13-17, 2002, Portland, Oregon, USA.

in any lexicon based system. Certain categories of words – morphologically derived words (plurals, varying verb conjugations, affixes, etc.) and proper nouns – are often not present in lexicons[7]. We were able to address the lack of plurals with reasonable success by using the Lingua-EN-Inflect Perl module[5] to create an auxiliary mapping from words to their plural forms. Some verb conjugations could be handled in a similar way, but generating proper nouns from the lexicon would be a difficult, if not futile, task. Expanding the lexicon could solve many of these problems. For example, a text containing many names of people and places, such as an encyclopedia, would greatly increase the recall of proper nouns. But the requirement that the lexicon contain a word's syllabication as well as the word itself makes this impractical. If we remove the syllabication requirement, we can enhance our lexicon substantially.

3. CORPUS BASED WORD ASSEMBLY

To circumvent the need for the syllabicated form of a word, we use a variant of the maximum matching algorithm[3] that keeps every match. This alleviates our dependence on having syllabication information and also removes the previous requirement that the input be in syllable form.

We created a much richer lexicon. Instead of using a wordlist of 100,000 words (about 1MB) from Webster's dictionary, we obtained every etext produced by Project Gutenberg[6] in the year 1999 – a 100MB collection that has 160,000 unique words. By allowing the use of diverse corpora, we can decrease the number of words not found in our lexicon. Instead of requiring that a word be a dictionary entry, now it need only be present in our corpus. This means we can create a suitable lexicon by selecting appropriate corpora and, as need arises, the lexicon can be easily augmented. Supporting a language requires only that we have a corpus in that language. In our case, we selected English corpora containing a variety of plural forms, verb conjugations, and proper nouns.

The algorithm proceeds iteratively through the input. If a string exists in the lexicon we consider it a word. When we find a word, we store it with the current syllable. For each syllable, we check if it is a word and step through each subsequent syllable, checking if the concatenation of preceding syllables is a word. We stop when the end of input is reached or a clue, such as punctuation, indicates a clear word boundary. The result is a set of possible words that correspond to each input syllable – just as we had in the syllabicated dictionary method.

4. TEST RESULTS

To evaluate these systems, we obtained the lyrics of five English songs in syllable form from Mutopia[8], comprising 1200 words after manual reconstruction. Of the possible results each system generated, we took the result with the fewest number of words and compared them based on precision and recall metrics. We define *precision* as the percent of returned words that occur in the manually reconstructed text in the same position and *recall* as the percent of words in the manually reconstructed text returned.[3]

For the syllabicated dictionary approach, we measure precision at 83.5% and recall at 77.5%. Using the corpus based method we achieve precision of 91.1% and recall of 86.1%. The corpus based method was also significantly faster, though there are pathological cases where it might not be.

5. FUTURE WORK

We intend to store all paths when the lyrics are reconstructed after OMR processing and develop a supporting XML-based document model. Doing so will increase recall at the expense of precision, but only until the correct path is determined. Our next step is to evaluate the multiple paths produced by the system. Ways to do this include manual selection and a word n-gram statistical model. Consistent with our philosophy, we plan to use a combination of these approaches. We will develop a statistical approach for ranking the available paths. This will allow collection managers to set probability thresholds above which the system could automatically select the "correct" path. Finally, we will create a web-based interface to allow others to experiment with this system.

6. CONCLUSIONS

This tool addresses an important issue for collection managers working to ingest musical scores that contain lyrics – bridging the gap between the way lyrics are written and the way they are searched. Collection managers will be able to enhance access by providing search access to full-text lyrics without manual transcription. Because the algorithm is simple and we plan to release it as open source software, other organizations will be able to experiment and adapt it to their needs.

7. ACKNOWLEDGMENTS

The Levy Project is funded through NSF's DLI-2 initiative (Award #9817430), an IMLS National Leadership Grant and support from the Levy Family.

We also thank the four anonymous reviewers for insightful comments.

8. REFERENCES

[1] Choudhury, G. S. *et al.* Digital Workflow Management: The Lester S. Levy Digitized Collection of Sheet Music. *First Monday*, 5(6), June 2000.

[2] Choudhury, G. S., *et al.* Strike Up the Score: Deriving Searchable and Playable Digital Formats from Sheet Music . *D-Lib Magazine*, 7(2), February 2001.

[3] D. D. Palmer. A trainable rule-based algorithm for word segmentation. In P. R. Cohen and W. Wahlster, editors, *Proceedings of the Thirty-Fifth Annual Meeting of the Association for Computational Linguistics and Eighth Conference of the European Chapter of the Association for Computational Linguistics*, pages 321–328, Somerset, New Jersey, 1997. Association for Computational Linguistics.

[4] N. Porter, editor. *Webster's Revised Unabridged Dictionary.* G and C. Merriam Co., 1913.

[5] Proc. Perl Conference 2.0. *An Algorithmic Approach to English Pluralization*, 1998.

[6] Project Gutenberg. *http://promo.net/pg/.*

[7] Sproat, R., Shih, C., Gale, W., Chang, N. A Stochastic Finite-State Word-Segmentation Algorithm for Chinese. *Computational Linguistics*, 22(3), 1996.

[8] The Mutopia Project. *http://www.mutopiaproject.org/.*