Automatic Harmonization Viewed as a Machine Translation Problem

Dept. of CIS - Senior Design 2014-2015*

Nicole Limtiaco
limni@seas.upenn.edu
Univ. of Pennsylvania
Philadelphia, PA

Rigel Swavely
rigel@seas.upenn.edu
Univ. of Pennsylvania
Philadelphia, PA

ABSTRACT

*The purpose of this document is three-fold. First, it describes the requirements of the project proposal. Second, it presents this information in a style your proposal should mimic. Third, by viewing the *.tex 'code behind', some basic LaTeX technique can be learned. Do not feel compelled to completely follow our style – but do use it as a guide.*

Like this 'report', your proposal is expected to have an abstract. An abstract is a two-paragraph maximum executive summary of your work. It should briefly outline your problem statement and its (expected) contributions.

1. INTRODUCTION

Many would consider music composition to be an art form that is accomplished primarily through human creativity. Writing music is a process that seems to require complex thought which fundamentally cannot be boiled down to a list of straightforward processes. Our project aims to answer the question, "Can computers imitate the complex and creative process of composing music?" We narrow the question down to the more specific and well-defined problem of generating harmony lines for a given melody line. Specifically, we would like to create 4-part chorale settings given the first part, the melody.

We will define a *melody* as a sequence of input *notes*, where each note contains information about pitch and timing. A *harmony* will be a sequence of output notes which are produced with the constraint that the sequence of notes support the input melody. We define the term *voice* to be some sequence of notes, either a melody or one of the harmonies, with a certain set of identifying characteristics. For example, the bass part in rock music is one type of voice and the soprano's part in an opera is another. The end result we wish to generate will be a group of 4 voices–the input melody and three automatically-composed harmonies–that, when played together, sound coherent and pleasant.

We propose to approach the problem from the view point of machine translation, where the input language is the melody and the target language is the specific harmony part you wish to generate. The first analogy that can be drawn between a pair of natural languages and a pair of melody and harmony voices is that both have a sense of "word translations". Just as there may be several words in the target language that could be sensible translations of a given word in the input langauge, there are also several notes in the harmony "language" that can harmonize well with an input melody note. Importantly, however, it is not the case that any note can harmonize well with the melody note. Some notes will sound dissonant when played together and still other notes may not even be in the harmony language, since harmony voices can have specified note ranges.

The second analogy that allows us to view this problem as a machine translation problem is that, like natural language, only certain strings of tokens (i.e. notes) are sensible in a given harmony voice. For example, the statement "colorless green ideas sleep furiously" contains all english words but is unlikely to be understood by an english speaker because the string of words in not sensible based on the rules of our language. Similarly, a very inharmonious sequence of notes may not sound sensible in the context of its harmony voice, if the notes are even recognized as music at all.

2. RELATED WORK

Automatic harmonization is a subset of the automatic musical composition problem, which dates as far back as the field of artificial intelligence itself. Perhaps the earliest work in automatic composition is Hiller and Isaacson's *Illiac Suite* [4], which is widely accepted as being the first musical piece composed by an electronic computer. Hiller and Isaacson used a generate-and-test method that generated musical phrases psuedo-randomly and kept only those that adhered to a set of music-theory-inspired heuristics.

Staying in line with the use of musical rules, Ebcioğlu [2] provided a break-through in the specific field of automatic harmonization through his CHORAL system. CHORAL, bolstered by about 350 musical rules expressed in first order logic, performed the task of writing four-part harmonies in the style J.S. Bach. With these logical predicates, Ebcioğlu reduced the problem of composing a harmony to a simple constraint satisfaction problem. Similar later works, notably by Tsang & Aitkin [9], also crafted the harmonization problem as a problem in constraint satisfaction, but with a significantly smaller amount ($\sim$20) of musical rules. The result of these constraint-based works were pieces of music that were non-offensive to the ear in that there were no dissonant musical patterns allowed by the constraints; however, creativity and overall similarity to human composition was still lacking.

More recent works began to put data-driven methods into use in order to create more human-like compositions. A simple case-based model implemented by Sabater *et al.* [6] was built to generate accompanying chords for a melody line.

*Advisor: Insup Lee (lee@cis.upenn.edu).

To a choose a harmony chord for a given context of previous harmony chords and the currently sounding melody note, the system would check a case base to see if any cases had the same harmony context and melody note and use the corresponding harmony chord if such a match were found. Musical heuristics were used to generate a chord if no match was found in the case base. An automatic harmonizer utilizing neural networks was also built by Hild *et al.* [3] to produce a harmony chord for a given melody quarter beat. Input features to the network included harmony context, current melody pitch, and whether or not the beat was stressed.

As these examples show, the previous harmony context and the melody pitch are important signals in deciding what the current harmony phrase should be. Many works have been conducted that model these signals using n-gram Markov Models. A Markov model assigns probabilities to some event C_t conditioned on a limited history of previous events $[C_{t-1}... C_{t-(n-1)}]$, implictly making the assumption that the event C_t depends only on a small amount of information about its context. For example, a system called MySong produced by Simon *et al.* [8] generates chord accompaniment given a vocalized melody by using a 2-gram Markov model for the harmony context and a 2-gram Markov model for the melody context. A similar system implented by Scholz *et al.* [7], which also generates chord accompanimants, experimented with 3-gram to 5-gram Markov models and incorporated smoothing techniques commonly seen in NLP to account for cases in whcih the model has not seen a context that is present in the test data. Most recently, Raczyński *et al.* [5] use discriminative Markov models that model harmony context, melody context, and additionaly the harmony relationship to the tonality.

The recent Senior Design Project by Cerny *et al.* [1] also used melody pitch and previous harmony context as their main signals for determining the next harmony chord to generate. However, they used these signals as inputs to an SVM classifier, as opposed to training a Markov model.

3. PROJECT PROPOSAL

Now is the time to introduce your proposed project in all of its glory. Admittedly, this is not the easiest since you probably have not done much actual research yet. Even so, setting and realizing realistic research goals is an important skill. Begin by summarizing what you are going to do and the expected benefit it will bring.

3.1 Anticipated Approach

Having summarized *what* you are going to do, its time to describe *how* you plan to do it. Our factorization example does not work so well here (it is likely impossible to realize) – so let us suppose you are going to create a service that takes a cell-phone picture of a building and returns via text-message, the name of that building[1].

In this case you might want to talk about establishing a server to receive pictures via MMS. Once the picture is received, you will run an edge extraction algorithm over it. Then, similarity between the submitted picture and those stored (and tagged) in a MySQL database will be computing using algorithm XYZ. Finally, the tag of the most similar image will be returned to the user. Do not bore the reader with trivial details, but give them an overview; a block-flow diagram would prove helpful (and is required).

[1]Do not use this idea – someone did it in a previous year.

3.2 Technical Challenges

In this subsection note where you anticipate having <u>novel</u> difficulty. Maybe you have never setup a MySQL database or even used SQL before at all – yes, that is a challenge – but not one readers care about. More novel would be the fact that many buildings on Penn's campus look similar and your classifier may be inaccurate in such instances. The purpose of this section is two-fold: 1) you will think about which parts of your project would require the most time and effort and 2) you will convince the reader that this is a project worth undertaking.

3.3 Evaluation Criteria

Suppose you have implemented your approach and it is functioning. Now how are you going to convince readers your approach is better than what exists? In the factorization example, you could just compare run-times between algorithms run on the same input. The image recognition example might use a percentage of accurate classifications. Other fields may have established testing benchmarks.

No matter the case, you need to prove you have contributed to the field. This will be easier for some than others. In particular, those with 'sensory' projects involving visual or sonic elements need to think this point through – objective measures are always better than subjective ones.

4. RESEARCH TIMELINE

We have completed some preliminary steps for our project and plan to complete our work along this proposed timeline.

- ALREADY COMPLETED: Preliminary reading. Experimented with musci21 library and corpus by implementing a deterministic harmony of thirds generator. Initial steps taken to train language model and translation models.
- PRIOR-TO THANKSGIVING : Finish training language model and translation model. Write beam search to find solution for one harmony voice.
- PRIOR-TO CHRISTMAS : Tune beam size and n-gram size. Implement code to generate harmony voices in different orders and determine which order is best.
- COMPLETION TASKS : Verify implementation is bug-free. Complete write-up. Get humans to evaluate output.
- IF THERE'S TIME : Investigate incorporating rhythm and timing decisions into the generated harmonies.

5. REFERENCES

[1] David Cerny, Jiten Suthar, and Israel Geselowitz. U-amp: User input based algorithmic music platform. 2014.

[2] Kemal Ebcioğlu. An expert system for harmonizing chorales in the style of j.s. bach. *The Journal of Logic Programming*, 8(1 - 2):145 – 185, 1990. Special Issue: Logic Programming Applications.

[3] Hermann Hild, Johannes Feulner, and Wolfram Menzel. Harmonet: A neural net for harmonizing chorales in the style of j. s. bach. In J.E. Moody, S.J. Hanson, and R.P. Lippmann, editors, *Advances in Neural Information*

Processing Systems 4, pages 267–274. Morgan-Kaufmann, 1992.

[4] L.A.J. Hiller and L.M. Isaacson. *Experimental Music: Composition with an Electronic Computer*. McGraw-Hill, 1959.

[5] Stanisław A. Raczyński, Satoru Fukayama, and Emmanuel Vincent. Melody harmonization with interpolated probabilistic models. *Journal of New Music Research*, 42(3):223–235, 2013.

[6] Jordi Sabater, J. L. Arcos, and R. López De Mántaras. Using rules to support case-based reasoning for harmonizing melodies. In *Multimodal Reasoning: Papers from the 1998 AAAI Spring Symposium (Technical Report, WS-98-04). Menlo Park, CA*, pages 147–151. AAAI Press, 1998.

[7] R. Scholz, E. Vincent, and F. Bimbot. Robust modeling of musical chord sequences using probabilistic n-grams. In *Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on*, pages 53–56, April 2009.

[8] Ian Simon, Dan Morris, and Sumit Basu. Mysong: Automatic accompaniment generation for vocal melodies. In *CHI 2008 Conference on Human Factors in Computing Systems*. Association for Computing Machinery, Inc., April 2008.

[9] C.P. Tsang and M. Aitken. Harmonizing music as a discipline of constraint logic programming. 1991.