Supporting Musical Creativity With Unsupervised Syntactic Parsing

Reid Swanson*†, Elaine Chew*‡ and Andrew S. Gordon*†

* University of Southern California Viterbi School of Engineering, Los Angeles, California
† University of Southern California Institute for Creative Technologies, Los Angeles, California
‡ Radcliffe Institute for Advanced Study at Harvard University, Cambridge, Massachusetts

Abstract

Music and language are two human activities that fit well with a traditional notion of creativity and are particularly suited to computational exploration. In this paper we will argue for the necessity of syntactic processing in musical applications. Unsupervised methods offer uniquely interesting approaches to supporting creativity. We will demonstrate using the Constituent Context Model that syntactic structure of musical melodies can be learned automatically without annotated training data. Using a corpus built from the Well Tempered Clavier by Bach we describe a simple classification experiment that shows the relative quality of the induced parse trees for musical melodies.

Introduction

Creativity is a difficult concept to define precisely, yet we all have an intuitive feeling for what is and is not creative. Although creativity is used to describe innovative and unique methods of accomplishing just about any task, the arts are most prototypically associated with creativity and the creative process. Music and language are two human activities that tie into this traditional notion of creativity well, and are particularly suited to computational exploration for several reasons. The ubiquity of these expressions and the ability for most people to have at least some limited experience or ability in these areas are key aspects that make these topics so appealing for research. Another reason is the relative ease in which basic units of "meaning" can be represented in various machine-readable formats. Language and music also share many characteristics that could allow key insights from one domain to shed light on the other.

There are many relationships that can be found between music and language dating as far back as Socrates, Plato and Aristotle. Dobrian (1992) elaborates three categories that are particularly recurrent in this discussion. The most relevant to this work is the concept of music as a language itself. When viewed in this way it is natural to apply linguistic theories of syntax and semantics to try to analyze and derive meaning from music. Dobrian (1992) ultimately argues that music is not a language in the same way English is, for example, because there are simply too many sonic elements that do not have a culturally defined meaning. However, even in this restricted view he believes that music contains many linguistic elements including symbols and grammar that allow linguistic analysis to be enlightening.

One of the more specific relationships between music and language is the natural ability to recursively group primitive elements together to form larger and larger units organized in a hierarchical structure. Syntax, or grammar, is a long and actively researched topic in the field of Linguistics that has been dedicated to these structures. Music does not have the rich history that Linguistics does in the area of syntax, but some work has been done, most notably by Lerdahl and Jackendoff (1983) and by Steedman (1996). There is even evidence that some of the syntactic processing for both types of data are processed in the same region of the brain (see, for example, Patel 2003), although not necessarily in the same way. Over the last decade syntactic processing has permeated nearly all aspects of natural language processing topics. Not only has the use of syntactic information directly improved the results of many traditional tasks, it has also enabled more complex systems to be built that were not possible without syntactic information.

Although, the use of syntactic information in the musical composition process and in music analysis is frequently discussed and often considered fundamental, there has been relatively little work on integrating this information into automated approaches to music analysis and generation. Two examples are the computer implementations of Lerdahl and Jackendoff's General Theory of Tonal Music (GTTM) by Hirata and Matsuda (2003), and by Hamanaka, Hirata, and Tojo (2006). However, due to ambiguities in the GTTM rules, the programs require human intervention to parse an input.

Hierarchical processing has also been integrated into some computing interfaces for music composition. Tuneblocks explicitly makes use of hierarchical units for teaching composition techniques to beginning and novice composers (Bamberger, 1999). Concepts of hierarchical construction have also been implemented in the *maquette* in OpenMusic[1]. Few automatic methods for generating music actually consider such hierarchical structures. Just as using syntactic representations has been useful in human

[1] http://recherche.ircam.fr/equipes/repmus/OpenMusic

guided learning processes, syntactic information could be useful for a host of computational music applications.

One starting point for developing automatic parsing techniques for music would be to adapt successful techniques from the language community to music. The most successful language parsing techniques require the development of large annotated corpora, which is riddled with difficulties. Early work in linguistic syntax analysis also showed that hand authored grammars are extremely difficult to create, debug and maintain. Until recently the quality of automatically learned grammars was found to be insufficient for even the most rudimentary tasks. Recent progress in unsupervised language parsing techniques have produced results that are much more competitive with state of the art systems, and may provide the ability to identify reasonable syntactic structure in music.

In this paper we argue for the necessity of syntactic processing in musical applications, particularly for two key aspects of creativity: enabling human creativity and being innately creative. We propose that unsupervised methods offer a uniquely interesting solution to both aspects. To demonstrate that these techniques are not just theoretically possible, we will describe how an unsupervised parsing technique developed by Klein and Manning (2002) for use with language can be used in the musical domain to parse musical melodies. In lieu of an annotated corpus of melodies, we describe a simple experiment that estimates the quality of the learned syntactic structures and shows their plausibility and promise for future research.

Computational Creativity

There are two main ways in which computers can play a significant role in the creative process of musical innovation. The first is by enabling the human to be more creative through facilitating mundane or arduous tasks not directly focused on the idea, yet are typically necessary for completing the process. For example, a musical score editor (e.g., Musicease[2] or Sibelius[3]) can greatly ease the creation and management of writing formal music notation. This can allow more focus and energy to be dedicated to the musical ideas and themes, in much the same way a word processor allows writers to spend more time on the content, and less time worrying about desktop publishing. The second way a computer can play a role in the creative process is for the computer itself to be creative. This second way is more alluring from an Artificial Intelligence point of view, however it is much more difficult to define exactly what it might mean.

Defining how a computer can be creative is a thorny topic not easy to resolve. However, there are at least two possibilities that are immediately apparent. One could determine the creativity of a system analogous to the Turing test by using human judges to compare the system's output to that of a human. Conversely, one could also determine the creativity of the system based on the process it uses to derive its output.

Having humans judge the level of creativity of a machine's output poses two significant theoretical problems. Although we would like to test how creative the machine is, by using humans we are in a sense testing how creative those people are in ascribing meaning to the output. Depending on the background of the person, their experiences and knowledge of the genre any computer-generated material may take on wildly different interpretations for different people, or even the same person in different circumstances. While this may be useful for inspiring new ideas in humans this is not adequately addressing the question "is the computer creative?" The other concern is epistemic. Even if our human judges were able to agree on an objective common knowledge for grading creativity, it would limit the performance of the machine to that of the human intellect. Although the human intellect is not likely to be the limiting constraint in the near future, it is not wise to build such a limitation into the definition of creativity. Finally, introspection about the creation should not just inspire us that it is good but *why* it is good, and afford the possibility of learning something beyond our own biases and capabilities.

Computational systems can embody these two types of creative processes in a multitude of ways. At one extreme are deterministic rule based systems, and completely unsupervised learning agents at the other end. Rule based systems are more naturally aligned with enabling humans to be more creative for several reasons. Often, as in the case of a musical score editor, the arduous tasks are well defined, and a series of rules can be written to alleviate much of the burden. Also, their behavior is typically expected or predictable, which can be beneficial for many applications, although the system's predictability usually subverts the possibility of its *being* creative. Rule based systems are not always predictable however. Wolfram tones[4] are a good example of how a seemingly simple rule can lead to unexpected emergent properties. Dabby (1996) also describes a music generation system that transforms human composed pieces in unpredictable and interesting ways using a chaotic mapping. By manipulating the initial conditions, one can produce variations that closely resemble the original piece or ones that deviate so much they become unrecognizable. These techniques can be used as novel methods of creating music but they are also useful in inspiring new ideas through the combination or recombination of sounds one may have never imagined.

Although the derivation of these deterministic systems is creative and they certainly inspire creativity in humans it is still difficult to say they are *being* creative. It seems more natural to align systems capable of learning with the act of being creative. Pachet's Continuator (2003) using Markov models, or the family of improvisation systems based on Factor Oracles described in Assayag and Dubnov (2004), Assayag et. al. (2006), and François, Chew and Thurmond

[2] http://www.musicease.com
[3] http://www.sibelius.com
[4] http://tones.wolfram.com/about/how.html

(2007) are recent examples of this type of model. Assayag et al. (2006) also describes a range of other musical improvisation systems based on statistical learning techniques. In these cases the act of creation is not the direct application of a known set of procedures, regardless of how unknown the output may appear, but is dependent on the input it is given and its ability to distinguish useful patterns.

Although rule based systems and learning systems tend to diverge in their respective strengths concerning the two types of computational creativity, there is no hard and fast line. For example *Voyager* (see George Lewis, 2000) is an early example, created in the 1980s, of a primarily rule based improvisation system that has 64 independent voices that can adapt to each other, and up to as many as two human performers. Similarly, learning based approaches often facilitate our own creativity, for example through simple actions like automatic text, or note, completion.

Our goal is to show that the unsupervised parsing methods described in this paper can fit within both aspects of computational creativity discussed in this section. The factor oracle and Markov models, as used in the improvisation systems mentioned above, can also be seen as an instance of unsupervised parsing, but without grammar induction. However, the use of these models has been motivated by their ability to train and execute in real time, and not necessarily for providing deep or structural analysis. Without these restrictions we will examine some other possible uses.

Parsing Music as Language

The widespread availability of massive amounts of computational power at all levels of devices from personal computers to embedded devices has produced an explosion of computational musical applications. These applications range from personal entertainment – Midomi[5] for music search, lastfm[6] for music recommendation, iTunes for organizing music – teaching and learning aids, to completely new methods of composing and generating music. All of these applications could be improved through the use of high-level musical understanding, which could facilitate the creative process for humans and machines.

For example, in the previous section the utility of a musical score editor was discussed. However, as more musical knowledge is available, even more intriguing applications become possible. With a little bit of linguistic knowledge, word processors are now armed with spell checkers, grammar checkers, thesauri, and a host of other tools that make them so much more than simple typesetting programs. Similarly, with more musical knowledge, a score editor could have the ability to highlight potential typos, identify irregularities in meter or rhythm, and suggest alternative notes and phrases. Another interesting application that comes closer to realization is completely automated music transcription. Unlike the score editor, which requires the person to have a certain level of proficiency in music notation, this type of system does not impose such a restriction. This could be a particularly useful training aid for someone who has picked up an instrument by ear, but has no formal training, for example.

Klapuri (2004) describes how such a system could be built using signal processing techniques (an acoustic model). As he notes, while these techniques produce desirable results they cannot account for the human listener's full experience or come close to the performance of a trained musician. One proposed method is to combine the acoustic model with a language model, analogous to most state of the art speech recognition systems.

The most prominently used models in speech recognition systems are n-grams (Jurafsky and Martin, 2000). These models estimate the probability of a word by conditioning on the words immediately preceding it, and combine these probabilities to estimate the probability of an entire sequence. Their popularity stems from their low complexity and relative high performance. Theoretically, however, they suffer from an inability to capture long distance dependencies in much the same way many of the improvisation systems do and will ultimately lead to a bottleneck in performance. The following example from Chelba and Jelenik (2000) highlights the issue:

> The contract ended with a loss of 7 cents **after** trading as low as 9 cents

Estimating the probability of the word *after* using a trigram model only conditions on the words *7 cents*. However, the words *7 cents* do not offer many clues that *after*, or any other word, is coming next. The subject *contract* and the verb *ended*, however, do seem to be better indicators. It is unlikely that a model, such as n-grams, that are based on word locality will ever be able to effectively capture this information. On the other hand, using a syntax-based model opens up the possibility of conditioning on syntactic locality, which has a much greater chance of leveraging long distance dependency relationships.

The development of the Penn Treebank (Marcus, 1993) has enabled the creation of high accuracy syntactic parsers for natural language processing tasks. Syntactic language models built from this data set have shown improvements over n-gram models in terms of perplexity, especially when the two are interpolated together (Chelba and Jelink 1998; Roark, 2000; Charniak 2001). In many areas these parsing techniques have already proven invaluable. Many of the top-performing question answering systems in the TREC competition (Dang et al., 2006), e.g. PowerAnswer 3 (Moldovon et al., 2006), make use of syntactic analysis to extract deeper semantic representations that have led to significant performance leads over other techniques. It is not hard to think of analogous musical applications where a richer analysis could be beneficial, such as melody based search, author detection, classification, phrase suggestions, and composition analysis.

[5] http://www.midomi.com
[6] http://www.last.fm

One of the biggest problems with these supervised parsing techniques is the lack of available training data. Although seemingly large, with over one million words, the Treebank only scratches the surface of what is needed to adequately cover the English language, let alone other languages. The situation only gets worse for music. Not only are there no available large corpora of syntactically annotated data but also the drop in performance going from one genre to another is likely to be even greater than, for example, moving from the Wall Street Journal text to fictional literature. Even if a suitable collection of genres could be identified developing annotation guidelines would be particularly difficult. Reaching a consensus for English, where the history of linguistic research dates back over 100 years, was arduous enough. For music, on the other hand, there has been even less debate on the appropriate formalisms for syntactic analysis, potentially making the development of a corpus even more difficult.

Although there is typically a severely limited supply of annotated data, there is usually an abundance of un-annotated data, in both language and music. In the absence of annotated data, supervised techniques, which are able to learn only from structured annotated data, are no longer feasible. Instead, unsupervised methods, which try to induce structure from un-annotated data, can be used. Although unsupervised methods generally have not performed as well as their supervised counterparts, they at least offer the possibility of some analysis, and the promise of improved performance in the future. At the very least they could be used to bootstrap the process of annotating data, where having partially annotated data to start with, even if incorrect, has shown to decrease the development time and increasing the overall accuracy (Marcus, 1993).

Constituent Context Model

Until fairly recently unsupervised parsing techniques had not been competitive with their supervised counterparts. In fact, unsupervised parsing has been such a difficult task that they have even had trouble surpassing the performance of a simple right branching rule baseline. Klein and Manning's Constituent Context Model (CCM) is the first such technique that showed significant improvements over this baseline, and is relatively competitive with supervised techniques, reaching an unlabelled F-score of over 70%. Although this is not necessarily the absolute best performing unsupervised system today, its performance is still near state-of-the-art, and is easy to adapt to new domains.

The Inside-Outside algorithm is one of the standard techniques used for grammar induction. One typically starts with a template grammar and uses the un-annotated data to iteratively learn probabilities for each rule by estimating the number of times that rule is used in all possible parse trees for all the sentences in the training data. The Inside-Outside algorithm suffers from several problems that have inhibited it from producing strong results. Two of the most prominent are that it is very sensitive to the initial parameters, and that the guaranteed increase in likelihood of the rules will not necessarily produce linguistically, or musically, motivated results.

The CCM model is a derivative of the Inside-Outside algorithm that attempts to address these two issues. The basic tenet of the CCM model is that there are two main properties that determine the constituency of a phrase:

1) the phrase itself, and
2) the context in which the phrase appears.

The model essentially gives up trying to find labeled rules that lead to a good derivation. Instead at every step in building the (binary) parse tree, it asks, "given the span of words dominated by this node and the context in which they are surrounded, is this span a constituent or not". Similar to the Inside-Outside algorithm it uses a dynamic programming approach to estimate the number of times each span of words and each context is seen.

Musical CCM Model

The CCM model works well for English, but does it work well for music? This is probably much too difficult a question to answer in general. However, if we limit the scope to melodies as a start, the model seems to make sense. As with English, melodic phrases are highly determined by the notes in the phrase segment itself, and the notes surrounding this segment. So, adapting the model from language to musical melodies should be relatively straightforward. By replacing words with melodic features one does not even need to make any underlying changes to the model. We explored this approach by inducing a grammar from a corpus of melodies using the CCM model. One of the major issues, however, is choosing the right melodic features to encode. In this section we discuss the corpus we developed, and how it was encoded.

Several considerations were made when compiling a corpus. The original corpus used by Klein and Manning contained approximately 7000 sentences. To ensure we had enough data to adequately train our model, we aimed to amass a corpus of equivalent size. Due to limitations of the CCM model, we chose to have phrases under 10 tokens in length. Additionally, as a first attempt we thought it prudent to choose a genre that was fairly well structured to give the model a good chance to succeed. For these reasons, we chose the fugues from Bach's Well-Tempered Clavier. All 48 Fugues are available in the kern machine-readable format from http://kern.humdrum.net. These work particularly well because fugues are made up of multiple independent voices that combine to form a single harmony. It is therefore possible to separate out each voice, and treat each one as a separate melody, and thus dramatically increase the amount of training data in the corpus. Since the voices essentially extend throughout the entire piece, a method for breaking them into shorter phrases was needed.

To segment the voices into phrases, we used Grouper, a publicly available segmentation algorithm developed by Temperley (2001), that has been shown to perform well compared with other automated segmentation algorithms

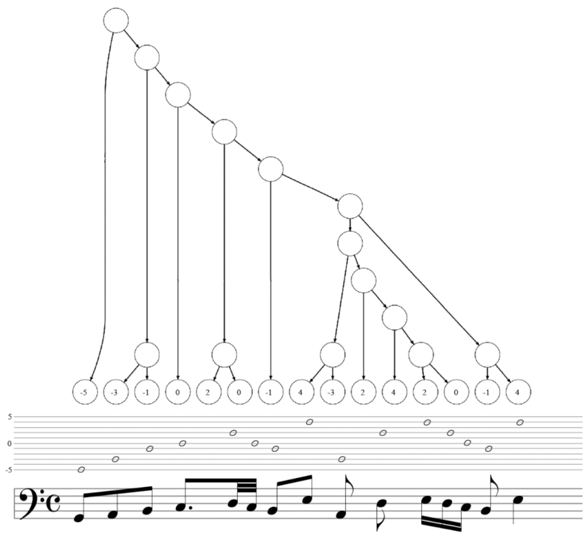

Figure 1: A sample melody from the Well-Tempered Clavier, Volume 1, Fugue 1 along with its parse generated using encoding (2) as seen above the musical notation.

(Thom, Spevak and Hoethker 2002). Grouper is available as part of Sleator and Temperley's Melisma program[7]. Although this program does not allow a hard limit on phrase length, which would not be appropriate in this case, the user can set the preferred length. The corpus, after applying Grouper, resulted in a collection of about 5000 melodies comprised of phrase segments approximately 10 notes in length.

The second major consideration for the corpus is the encoding to use. If the encoding is too fine grained, then the small amount of training data will pose a problem; on the other hand, if the representation is too coarse, then there will not be sufficient information from which to learn distinguishing cases. In language, part of speech tags are often used as a compromise. In music, we deal with a similar problem. With even the most naïve view of a melody, each basic unit (the note) comprises of at least a pitch and an associated duration. Even if we quantize pitch values, there is still a wide range of possible values for most instruments; the same holds true for possible duration values. Since Grouper requires that the data be encoded as triplets of onset, offset and a midi pitch value, we chose to examine the following six possible encodings based on these values:

1) absolute pitch value (midi value from 0 to 127),
2) pitch value relative to key,
3) first pitch relative to key, others relative to prior,
4) absolute duration (offset – onset),
5) duration relative to average, and
6) first duration relative to average, others relative to previous duration.

[7] http://www.link.cs.cmu.edu/music-analysis

We used the spiral array center of effect generator key finding algorithm developed by Chew (2000) to locate the key of each melody, after applying a pitch spelling algorithm (Chew and Chen 2005), for the second encoding. Since the primary concern of a melody is the tune, and usually not the octave in which it is played, we shifted the key to the octave of the first note in the melody. This key is then mapped back into the appropriate midi value. Although some information is lost, the benefits seem to outweigh the consequences. One can devise other combinations of the pitch and duration information that would also be natural, but have not been tried at this point. Figure 1 illustrates an example of an example melody using the second encoding. Just above the melody is a visual representation of the encoding with the notes positioned on a graph base on how far from the key they are. The numeric value on the graph represents the distance in midi tones the pitch is from the shifted key. Above the encoding is a sample parse tree output from the system.

Experiments

Since there are no readily available annotated corpora to evaluate the quality of the melodic parses, a method for determining the quality of the trees was necessary. Building an evaluation corpus was one option. Due to the cost of development, and because it is delicate to create your own evaluation corpus when the question of bias may taint the results regardless of its integrity we chose not to pursue this option. Perplexity is a metric that is often used for evaluating the quality of a language model. However, this was not possible because, unlike the traditional Inside-Outside algorithm, the CCM model does not generate true probabilities.

To try to measure a similar predictive quality that perplexity captures, we devised a simple classification experiment. For each melodic phrase in the test corpus, another melody with the same symbols, but reordered randomly, was created. The trained model was then used to choose which sequence in the held out test corpus was more likely to be a melody from a Bach Fugue. Using these guidelines a 20-fold cross validation experiment was run for each of the six encodings.

The results are summarized in Table 1. As can be expected using the absolute values for pitch and duration are too fine grained, and do not lead to the best results. The best results were achieved using the relative duration encoding (type 6). Each of the pitch encodings performed roughly equivalently, although normalizing to the key did produce slightly higher scores on average. In all cases the performance was well above a 50% baseline, showing that there is enough information from which to learn, and that the model is able to capture at least some of that information. The variation in performance indicates that the encoding is an important factor for high performance classification. While the performance metric suggests that melodic parses based on relative duration are the most predictive and well formed, it makes no guarantee that

Feature Encoding	% Correct
1) absolute pitch	83.36 ± 1.03
2) pitch relative to key	84.29 ± 1.24
3) pitch relative to previous	83.14 ± 1.00
4) absolute duration	72.95 ± 2.78
5) duration relative to average	70.62 ± 1.41
6) duration relative to previous	90.69 ± 1.17

Table 1: The percentage of correct classifications between real and randomly shuffled data for each of the six encodings.

these parses are the most theoretically or musically interesting.

It is probable another more sophisticated classification algorithm such as Maximum Entropy, Support Vector Machines or even an n-gram language model could perform better at this task. Our goal is simply to show theoretically the plausibility of these abstracted tree structures. Many of the melodic phrase encoding have little or no variation because the same duration is repeated a significant number of times, for example. In these cases the classification was considered incorrect because neither sequence was considered more likely and leads to an upper-bound in performance that is less than 100%. Regardless of improvements in the encoding or using other classification algorithms there is relatively little room for performance increases above the best encoding because of this upper-bound.

Future Work

These initial results are encouraging and suggest that the CCM model is able to learn an adequate grammar for musical melodies. There are still several open questions that we would like to explore however. The encodings we have chosen are only a few of the possibilities and it would be interesting to experiment with more complex combinations. Our encoding also chooses the key with regard to the local phrase segment, but another valid option would be to use the global key from the entire piece.

The CCM model is not the only unsupervised parsing model available, and it would be interesting to see the results of other techniques. For example, Bod (2001) applied his Data Oriented Parsing model to the task of melodic segmentation. Since then, he has adapted his model specifically for unsupervised parsing and has shown highly competitive results (2006). It might be worth considering Bod's model for the specific task of musical grammar induction as well.

From a practical standpoint a more indicative test of any supervised or unsupervised parsing model will be how it impacts system performance in real world tasks, such as the value added (or not) when such models are integrated into an automated transcription program. An alternatively good test might be the integration in a computer-assisted composition tool for use as a visual aid. Experts could then qualitatively decide whether having the tree structure information available is useful without having to do full evaluations using an annotated corpus. From a more theoretical viewpoint it would also be interesting to have experts rate the quality of the melodic parses in some way, either through a gold standard or through qualitative ratings.

Despite the theoretical benefits of parsing, and especially unsupervised parsing, there remain many drawbacks and opportunities for further research. Although indirectly useful through enabling and improving new applications such as question answering, there are few definitive results that successfully integrate syntactic language models, supervised or not, directly into an application framework. Although unsupervised models have the potential to match the performance of supervised ones, they still lag far behind supervised models.

In summary, syntactic analysis has the potential to change the landscape of musical applications, the same way high accuracy parsing has enabled a variety of applications previously unimaginable in the language processing community. While the Natural Language Processing community has benefited from the Penn Treebank, there is no analogous resource available from music. Developing such a corpus is likely to be fraught with even more challenges for deciding which formalisms to use, and genres to annotate. Unsupervised parsing techniques offer a way to address this issue because they do not require any annotated data. These techniques offer the possibility of learning something entirely new because they are not limited by any particular formalism, nor are they bound by the noise introduced by inter-rater disagreements. This combination of characteristics addresses both key aspects of creativity by enabling computers to assist in more complex tasks, and by imparting the ability to learn from data beyond a particular rule set.

Acknowledgements

The project or effort described here has been sponsored by the U.S. Army Research, Development, and Engineering Command (RDECOM). Statements and opinions expressed do not necessarily reflect the position or the policy of the United States Government, and no official endorsement should be inferred.

References

Assayag, G., Bloch, G., Chemillier, M., Cont, A., Dubnov, S. 2006. OMax Brothers: A Dynamic Topology of Agents for Improvization Learning. In Proceedings of the 1st ACM workshop on Audio and music computing multimedia, 125–132.

Assayag, G., Dubnov, S. 2004. Using Factor Oracles for Machine Improvization, G. Assayag, V. Cafagna, M. Chemillier, eds., Formal Systems and Music special issue, Soft Computing 8, 1432–7643.

Bamberger, 2000. *Developing Musical Intuitions*. Cambridge, Mass.: MIT Press.

Bod, R. 2001. Stochastic Models of Melodic Analysis: Challenging the Gestalt Principles Proceedings 14th Meeting of the FWO Research Society on Foundations of Systematic Musicology Ghent, Belgium.

Bod, R. 2007. An All-Subtrees Approach to Unsupervised Parsing. Proceedings ACL-COLING 2006, Sydney.

Charniak, E. Immediate Head Parsing for Language Models. In Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics, 2001.

Chelba, C. and Jenilek, F. 1998. Exploiting syntactic structure for natural language modeling. In proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and the 17th International Conference on Computational Linguistics and 17th International Conference on Computational Linguistics, 225-231.

Chelba, C. and Jenilek, F. 2000. Structured language modeling. Computer Speech and Language, 14, 283–332.

Chew, E. 2000. *Towards a Mathematical Model of Tonality*. Ph.D. dissertation. Operations Research Center, MIT. Cambridge, Mass.

Chew, E., and Chen, Y.-C. (2005). Real-Time Pitch Spelling Using the Spiral Array. *Computer Music Journal*, 29(2), pp.61-76.

Dabby, D. S. 1996. Musical variations from a chaotic mapping. Chaos: An Interdisciplinary Journal of Nonlinear Science, June 1996, Vol.6, Iss.2, pp.95-107.

Dang H. T., Lin, J. and Delly, D. 2006. Overview of the TREC 2006 Question Answering Track. In proceedings of the Fifiteenth Text Retrieval Conference.

Dobrian, C. 1992. Music and Language. Available at http://music.arts.uci.edu/dobrian/CD.music.lang.htm.

Hamanaka, M., Hirata, K., and Tojo, S. 2006. Implementing "A Generative Theory of Tonal Music". *Journal of New Music Research*, 35(4), 249–277.

Hirata, K., and S. Matsuda. 2003. Interactive Music Summarization based on Generative Theory of Tonal Music. *Computer Music Journal* 27(3), 73–89.

Jurafsky, D., and Martin, J. 2000. Speech and Language Processing: An Introduction to Natural Language Processing, Speech Recognition, and Computational Linguistics. Prentice-Hall.

Klapuri, A. " Signal processing methods for the automatic transcription of music," Ph.D. thesis, Tampere University of Technology, Finland, April 2004.

Klein, D. and Manning, C. D. 2002. A generative constituent-context model for improved grammar induction. Proceedings of 40th Annual Meeting of the Assoc. for Computational Linguistics.

Lerdahl, F., and Jackendoff, R. 1983. *A Generative Theory of Tonal Music*. Cambridge, Mass.: MIT Press.

Lewis, G. 2000. Too Many Notes: Computers, Complexity and Culture in *Voyager*, *Leonardo Music Journal* 10, 33–39.

Marcus, M. P., Marcinkiewicz, M. A., and Santorini, B. 1993. Building a large annotated corpus of English: the penn treebank. Comput. Linguist. 19, 2 (Jun. 1993), 313-330.

Moldovan, D., Bowden, M., Tatu, M. 2006. A Temporally-Enhanced PowerAnswer in TREC 2006. In proceedings of the Fifiteenth Text Retrieval Conference.

Pachet, F. 2003. The Continuator: Musical Interaction With Style. *J. of New Music Research*, 32(3), 333–341.

Patel A.D. (2003). Language, music, syntax and the brain. Nature Neuroscience 6(7):674-681.

Roark, B. 2001. "Probabilistic top-down parsing and language modeling," Computational Linguistics, vol. 27, no. 3, pp. 249--276.

Steedman, M. 1996. The Blues and the Abstract Truth: Music and Mental Models, In A. Garnham and J. Oakhill, eds., *Mental Models In Cognitive Science*. Mahwah, NJ: Erlbaum, 305–318.

Temperley, D. 2001. *The Cognition of Basic Musical Structures*. Cambridge, Mass.: MIT Press.

Thom, B., Spevak, C., Hoethker, K. 2002. Melodic Segmentation: Evaluating the Performance of Algorithms and Musical Experts. In *Proceedings of the International Computer Music Conference*.