We contrast two opposing approaches to building bots that autonomously learn to rap battle: a symbolic probabilistic approach based on induction of stochastic transduction grammars, versus a neural network approach based on backpropagation through unconventional transduction recursive auto-associative memory (TRAAM) models. Rap battling is modeled as a quasi-translation problem, in which an appropriate output response must be improvised given any input challenge line of lyrics. Both approaches attempt to tackle the difficult problem of compositionality: for any challenge line, constructing a good response requires making salient associations while satisfying contextual preferences at many different, overlapping levels of granularity between the challenge and response lines. The contextual preferences include fluency, partial metrical or syntactic parallelism, and rhyming at various points across the lines. During both the learning and improvisation stages, the symbolic approach attempts to explicitly enumerate as many hypotheses as possible, whereas the neural approach attempts to evolve vector representations that better implicitly generalize over soft regions or neighborhoods of hypotheses. The brute force symbolic approach is more precise, but quickly generates combinatorial numbers of hypotheses when searching for generalizations. The distributed vector based neural approach can more easily confuse hypotheses, but maintains a constant level of complexity while retaining its implicit generalization bias. We contrast both the theoretical formulation and experimental outputs of the two approaches.

1. INTRODUCTION

Despite its status as one of the most influential developments in the recent history of music, rap and hip hop remains surprisingly underexplored in computer music. This may be ascribed in part to the extraordinary level of difficulty of the tasks involved in rapping. Perhaps the most difficult form of this genre is rap battling, in which a rapper must improvise on-the-fly responses to any challenge rap issued by another rapper.

Consider the many complex factors a rapper must integrate in constructing line 2 as a response, if given the line 1 as a challenge, in the following raps drawn from “The Magic Number” by De La Soul:

1: focus is formed by flaunts to the soul, souls who flaunt styles gain praises by pounds
2: common are speakers who are never scrolls, scrolls written daily creates a new sound

Some of the many complex factors the rapper would face:

- the response line should somehow be salient to the challenge line
- some phrases within the response line can somehow be salient to corresponding phrases within the challenge line—e.g., ‘focus is formed by flaunts to the soul’ is salient to ‘common are speakers who are never scrolls’
- some individual words within the response line can somehow be salient to corresponding words within the challenge line—e.g., ‘is’ is salient to ‘are’, and ‘who flaunt styles’ is salient in a different way to ‘written daily’
- the response line should flow fluently (yet sometimes may allow for stylistic ungrammaticality, disfluencies such as stuttering, or slang constructs)
- some phrases within the response line can use metrical parallelism to corresponding phrases within the challenge line—e.g., ‘scrolls written daily creates a new sound’ has a close meter to ‘souls who flaunt styles gain praises by pounds’
- some phrases within the response line can use syntactic parallelism to corresponding phrases within the challenge line—e.g., ‘focus is …’ is syntactically parallel to ‘common are …’
- the response line should typically rhyme with the challenge line—e.g., ‘pounds’ rhymes with ‘sound’
- some words or phrases within the response line may also be made to rhyme with the challenge line—e.g., ‘soul’ rhymes with ‘scrolls’, and ‘gain praises’ rhymes with ‘creates’
None of these are hard and fast rules or constraints; rather,
all the factors are merely soft biases or preferences. More-
over, each choice influences the other choices that need to be
made. The combinatorial context dependencies that thus arise
make computational complexity a severe issue for automatic
improvisation of rap responses.

To model the relationships between the challenge and re-
sponse at all these various levels of granularity, it is neces-
sary to take all the associated fragments of the two lines, and
compose them hierarchically into the full challenge-response
pair of lines. This gives a compositional relationship that can
be thought of as a tree whose leaves are the individual words
or phrases associated with each other by dint of salience, syn-
tactic function, or rhyme, and whose internal nodes are pro-
gressively longer compositions of the shorter chunks:

\[
[ [[ ‘focus’/’common’
   [ ‘is’/’are’
   [ [ ‘formed by’/’speakers’ ‘to the’/’who are never’
      ‘soul’/’scrolls’ ]] ]
   [ ‘/’/ ]
[ [[ ‘soul’/’scrolls’
   ‘who flaunt styles’/’written daily’ ]
   [ ‘gain praises’/’creates’
   [ ‘by’/’a new’ ‘pounds’/’sound’ ]] ]
\]

Such trees are highly reminiscent of bilingual parse trees (or
biparse trees) in machine translation. We can think of rap
battle improvisation as a quasi-translation task in which chal-
 lenges are “translated” into responses—not translation in the
conventional sense, but still, a task of relating one’s response
to any given challenge.

Computational complexity, as mentioned, is a major issue
for rap battle improvisation algorithms. But it becomes an
even more challenging problem for the task of automatically
learning the improvisation model, in ways that learn the im-
portant abstractions and generalizations. We contrast in this
paper two very different approaches to tackling the complex-
ity issues in learning compositional models for rap battle bots:
traditional symbolic approaches based on (a) probabilistic in-
duction of stochastic transduction grammars, versus (b) neu-
ral network approaches based on backpropagation training of
transduction recursive auto-associative memories. We con-
trast these two approaches in terms of, in turn, their represen-
tation, learning, improvisation, and empirical aspects.

2. SYMBOLIC VS. NEURAL COMPOSITIONAL
REPRESENTATIONS

2.1 Symbolic transduction grammar representations

The symbolic rap battle learning approach introduced by Wu
et al.[1] explicitly represents individual bilingual parse trees
like the one above, using stochastic versions of the syntax
directed transduction grammars (SDTGs) of classic formal
language theory [2]. The model restricts induction to the sub-
class of SDTGs known as inversion transduction grammars or
ITGs [3], for which polynomial time learning and prediction
algorithms exist (unlike SDTGs), and yet which have been
empirically demonstrated to possess attractive language uni-
versal properties for machine translation [4].

Rules (and instances of rules) represent structured correla-
tions between the input challenge language and output re-
sponse language. Formally, an ITG is a tuple \((N, \Sigma, \Delta, R, S)\),
where \(N\) is a finite nonempty set of nonterminal symbols, \(\Sigma\) is
a finite set of terminal symbols in \(L_0\) (output language), \(\Delta\) is
a finite set of terminal symbols in \(L_1\) (input language), \(R\) is a
finite nonempty set of inversion transduction rules and \(S \in N\)
is a designated start symbol. A normal-form ITG consists of
rules in one of the following four forms:

\[
S \to A, A \to [BC], A \to (BC), A \to e/f
\]

where \(S \in N\) is the start symbol, \(A, B, C \in N\) are nonter-
mental symbols and \(e/f\) is a biterminal. A biterminal is a pair
of symbol strings: \(\Sigma^* \times \Delta^*\), where at least one of the strings
has to be nonempty. The square and angled brackets signal
straight and inverted order respectively. With straight order,
both the \(L_0\) and \(L_1\) productions are generated left-to-right,
but with inverted order, the \(L_1\) production is generated right-
to-left.

Given a pair of input and output sentences \(e_1, \ldots, e_T\) and
\(f_1, \ldots, f_V\), respectively, an ITG generates a biparse tree by
recursively combining smaller bispans (chunks of aligned in-
put and output segments) into larger bispans using the synt-
tactic rules in straight or inverted order. Each bispans corre-
sponds to at least one nonterminal and is represented using a
4-tuple \(s, t, u, v\) which corresponds to the input segment with
tokens \(e_s, e_{s+1}, \ldots, e_t\) and the output segment with tokens
\(f_u, f_{u+1}, \ldots, f_v\).

In this symbolic approach, sets of biparse trees are repre-
seented explicitly as well, but for efficiency, tabular and hyper-
graph data structures are used wherever possible to compress
the storage of biparse trees that share subtrees (commonly re-
ferred to as charts or packed forests). Even so, because of the
large number of choices at each level of granularity, it is still
impractical to store anywhere near an exhaustive catalog of
improvisation hypotheses.

2.2 Neural transduction RAAM vector representations

An alternative approach that aims to reduce the need to ex-
plitly represent enormous numbers of similar competing hy-
potheses is to instead represent rap battle improvisation hy-
potheses using fixed-dimensionality continuous vectors, em-
ploying the new TRAAM (transduction RAAM) model of
machine translation proposed by Addanki and Wu [5]. The
distributed vector representations in TRAAM attempt to roughly
parallel the structural composition of a syntax directed trans-
duction grammar. However, unlike symbolic transduction
grammar based representations, the continuous vector rep-
resentations in effect represent soft neighborhoods of cross-
lingual associations. TRAAM implicitly learns context-sensitive
generalizations over the structural relationships, between the
corresponding parts of the challenges and responses across all
levels of granularity, while avoiding incurring the symbolic
models’ exponential cost of modeling context sensitivity.
More formally, TRAAM is a bilingual generalization of the way that the RAAM (recursive autoassociative memory model) of Pollack [6] monolingually approximates context-free grammars. In TRAAM’s distributed representation of an ITG, each bispan \( s, t, u, v \) is represented using a feature vector \( v_{stu} \) of dimension \( d \) which represents a fuzzy encoding of all the non-terminals that could generate the bispan. This stands in contrast to the symbolic ITG where each nonterminal that generates the bispan must be enumerated separately. As with symbolic ITGs, vectors corresponding to larger bispans are recursively generated from the vectors representing smaller bispans, but in TRAAM this is done using a compressor network. The compressor network takes two vectors of dimension \( d \), along with a single bit corresponding to straight or inverted order, and outputs a vector of dimension \( d \) — essentially compressing an input of \( 2d + 1 \) dimensions to a vector of dimension \( d \).

The role of the compressor network is analogous to the transduction rules in the ITG model, but with the important distinction of (1) keeping the encoding fuzzy, and (2) forcing generalization over similar vectors in the Euclidean space neighborhood. Figure 1 visualizes how transduction rule instances (both straight and inverted) correspond to inputs to the compressor network. Each nonterminal in an ITG can be encoded as a bit vector, identical to the vector of the bispan in our model. Using the universal approximation theorem of neural networks [7], an encoder with a single hidden layer can represent any set of transduction rules. Conversely, any variant of our model can be represented as an ITG by assuming a unique nonterminal label for the feature vector corresponding to each bispan. Hence, symbolic ITGs and neural TRAAMs represent two ways to model compositional bilingual relations. TRAAM’s neural encoding of nonterminals is better suited for modeling generalizations over bilingual relations without exploring the search space, while symbolic ITG representations avoid potential confusions due to accidental similarities between vectors.

3. SYMBOLIC VS. NEURAL RAP BATTLES

We now discuss runs of the symbolic versus neural models on actual data. Freely available user generated hip hop lyrics from the Internet were used as training data for our experiments. After minor preprocessing, the corpus contained 22 million tokens, comprising 260,000 verses, or 2.7 million lines. As human evaluation using expert hip hop listeners is expensive, a small subset of 85 lines was chosen as the test set to provide challenges for comparing the quality of responses generated by different systems.

### 3.1 Bilingual recursive neural network model

We use the bilingual recursive neural network model discussed earlier along with a token based transduction grammar model trained on around 200,000 lines of challenge response pairs. The challenge response pairs were selected using a rhyme scheme detection module proposed in Addanki and Wu [8]. We use the translation lexicon from the trained transduction grammar and use that along with the biparses to train our neural network model. Both these models are then used to improvise the responses using a 4-gram language model which was trained on the entire training corpus using SRILM [9]. The weights of the feature scores were determined empirically observing the performance on a small subset of the training data. In order to evaluate the performance of an out-of-the-box phrase-based SMT (PBSMT) system toward this novel task of generating rhyming and fluent responses, a standard Moses baseline [10] was also trained in order to compare its performance with our transduction grammar induction model.

### 3.2 Phrase-based SMT baseline performs poorly

Table 1 shows the average fraction of sentences rated good and acceptable for each model. Our bilingual neural network based model produces significantly higher percentage of good and acceptable rhyming responses compared to the phrase-based SMT (PBSMT) baseline. It is surprising that despite being a token based model, our model outperforms the segmental PBSMT model even on the criterion of fluency. These results indicate that our bilingual neural network model captures enough context to generate fluent responses, significantly augmenting the performance of a token based model.

### 3.3 Challenge-response examples

Table 2 shows some of the challenges and the corresponding responses of our model and the PBSMT baseline. It is interesting to note that our model produces responses comparable in fluency to PBSMT despite being a token based transduction grammar. However, PBSMT models tend to produce responses that are too similar to the challenge compared to the our model which improvise responses that rhyme better (shown in boldface). In fact our model frequently produces responses that rhyme words not only at the end but also in the
middle of challenges as our transduction grammar model captures structural associations more effectively than the phrase-based model.

4. CONCLUSION AND FUTURE DIRECTIONS

Teaching machines to rap battle is a quest that encapsulates numerous interacting levels of improvisational artistry in a complex, structured AI learning challenge. We have described an unconventional line of attack in which a recursive bilingual neural network sidesteps the exponentially complex hypothesis space needed by existing suitable symbolic learning models for both the improvisational response generation search and the model learning search, by instead using compositional distributed vector representations in which a single vector implicitly represents an entire neighborhood of multiple similar association patterns between corresponding structural aspects of challenges and responses. The fact that challenge-response association patterns that are structurally similar tend to have similar vectors allows training to learn soft, context-sensitive generalizations over all kinds of structural challenge-response associations patterns, from concrete to abstract patterns, and from short to long patterns.

Our approach is unlike conventional approaches to poetry in being completely unsupervised, making zero use of any linguistic or phonetic features in spite of an extremely unstructured and noisy domain. Modeling improvisation as a quasi-translation learning problem means that for any challenge, the machine must learn on its own what kinds of improvised responses would be fluent, salient, rhyming, and of similar metrical and syntactic structure. The distributed feature vectors that encode challenge-response association patterns are learned simultaneously by our recursive bilingual neural network, using context from both the challenge and the response. The soft structural relationships learned are used to improve the probabilistic responses generated by our improvisational response component, as judged by human rap listeners.

Acknowledgments

This material is based upon work supported in part by the Defense Advanced Research Projects Agency (DARPA) under BOLT contract nos. HR0011-12-C-0014 and HR0011-12-C-0016, and GALE contract nos. HR0011-06-C-0022 and HR0011-06-C-0023; by the European Union under the FP7 grant agreement no. 287658; and by the Hong Kong Research Grants Council (RGC) research grants GRF620811, GRF621008, GRF612806, FSGRF13E28, and FS-GRF14E4G35. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of DARPA, the EU, or RGC.

5. REFERENCES


