Learning to Freestyle: Hip Hop Challenge-Response Induction via Transduction Rule Segmentation

Dekai Wu Karteek Addanki Markus Saers Meriem Beloucif
Human Language Technology Center
Department of Computer Science
HKUST, Clear Water Bay, Hong Kong
{dekai|vskaddanki|masaers|mbeloucif}@cs.ust.hk

Abstract

We present a novel model, Freestyle, that learns to improvise rhyming and fluent responses upon being challenged with a line of hip hop lyrics, by combining both bottom-up token based rule induction and top-down rule segmentation strategies to learn a stochastic transduction grammar that simultaneously learns both phrasing and rhyming associations. In this attack on the woefully under-explored natural language genre of music lyrics, we exploit a strictly unsupervised transduction grammar induction approach. Our task is particularly ambitious in that no use of any *a priori* linguistic or phonetic information is allowed, even though the domain of hip hop lyrics is particularly noisy and unstructured. We evaluate the performance of the learned model against a model learned only using the more conventional bottom-up token based rule induction, and demonstrate the superiority of our combined token based and rule segmentation induction method toward generating higher quality improvised responses, measured on fluency and rhyming criteria as judged by human evaluators. To highlight some of the inherent challenges in adapting other algorithms to this novel task, we also compare the quality of the responses generated by our model to those generated by an out-of-the-box phrase based SMT system. We tackle the challenge of selecting appropriate training data for our task via a dedicated rhyme scheme detection module, which is also acquired via unsupervised learning and report improved quality of the generated responses. Finally, we report results with Maghrebi French hip hop lyrics indicating that our model performs surprisingly well with no special adaptation to other languages.

1 Introduction

The genre of lyrics in music has been severely understudied from the perspective of computational linguistics despite being a form of language that has perhaps had the most impact across almost all human cultures. With the motivation of spurring further research in this genre, we apply stochastic transduction grammar induction algorithms to address some of the modeling issues in song lyrics. An ideal starting point for this investigation is hip hop, a genre that emphasizes rapping, spoken or chanted rhyming lyrics against strong beats or simple melodies. Hip hop lyrics, in contrast to poetry and other genres of music, present a significant number of challenges for learning as it lacks well-defined structure in terms of rhyme scheme, meter, or overall meaning making it an interesting genre to bring to light some of the less studied modeling issues.

The domain of hip hop lyrics is particularly unstructured when compared to classical poetry, a domain on which statistical methods have been applied in the past. Hip hop lyrics are unstructured in the sense that a very high degree of variation is permitted in the meter of the lyrics, and large amounts of colloquial vocabulary and slang from the subculture are employed. The variance in the permitted meter makes it hard to make any assumptions about the stress patterns of verses in order to identify the rhyming words used when generating output. The broad range of unorthodox vocabulary used in hip hop make it difficult to use off-the-shelf NLP tools for doing phonological and/or morphological analysis. These problems are further exacerbated by differences in intonation of the same word and lack of robust transcription (Liberman, 2010).

Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, pages 102–112,
Seattle, Washington, USA, 18-21 October 2013.

We argue that stochastic transduction grammars,[1] given their success in the area of machine translation and efficient unsupervised learning algorithms, are ideal for capturing the structural relationship between lyrics. Hence, our FREESTYLE system models the problem of improvising a rhyming response given any hip hop lyric challenge as transducing a challenge line into a rhyming response. We use a stochastic transduction grammar induced in a completely unsupervised fashion using a combination of token based rule induction and segmenting (Saers *et al.*, 2013) as the underlying model to fully-automatically learn a challenge-response system and compare its performance against a simpler token based transduction grammar model. Both our models are completely unsupervised and use no prior phonetic or linguistic knowledge whatsoever despite the highly unstructured and noisy domain.

We believe that the challenge-response system based on an interpolated combination of token based rule induction and rule segmenting transduction grammars will generate more fluent and rhyming responses compared to one based on token based transduction grammars models. This is based on the observation that token based transduction grammars suffer from a lack of fluency; a consequence of the degree of expressivity they permit. Therefore, as a principal part of our investigation we compare the quality of responses generated using a combination of token based rule induction and top-down rule segmenting transduction grammars to those generated by pure token based transduction grammars.

We also hypothesize that in order to generate fluent and rhyming responses, it is not sufficient to train the transduction grammars on all adjacent lines of a hip hop verse. Therefore, we propose a data selection scheme using a rhyme scheme detector acquired through unsupervised learning to generate the training data for the challenge-response systems. The rhyme scheme detector segments each verse of a hip hop song into stanzas and identifies the lines in each stanza that rhyme with each other which are then added as training instances. We demonstrate the superiority of our training data selection method by comparing the quality of the responses generated by the models trained on data selected with and without using the rhyme scheme detector.

Unlike conventional spoken and written language, disfluencies and backing vocals[2] occur very frequently in the domain of hip hop lyrics which affect the performance of NLP models designed for processing well-formed sentences. We propose two strategies to mitigate the effect of disfluencies on our model performance and compare their efficacy using human evaluations. Finally, in order to illustrate the challenges faced by other NLP algorithms, we contrast the performance of our model against a conventional, widely used phrase-based SMT model.

A brief terminological note: "stanza" and "verse" are frequently confused and sometimes conflated. Worse yet, their usage for song lyrics is often contradictory to that for poetry. To avoid ambiguity we consistently follow these technical definitions for segments in decreasing size of granularity:

verse a large unit of a song's lyrics. A song typically contains several verses interspersed with choruses. In the present work, we do not differentiate choruses from verses. In song lyrics, a verse is most commonly represented as a separate paragraph.

stanza a segment within a verse which has a meter and rhyme scheme. Stanzas often consist of 2, 3, or 4 lines, but stanzas of more lines are also common. Particularly in hip hop, a single verse often contains many stanzas with different rhyme schemes and meters.

line a segment within a stanza consisting of a single line. In poetry, strictly speaking this would be called a "verse", which however conflicts with the conventional use of "verse" in song lyrics.

In Section 2, we discuss some of the previous work that applies statistical NLP methods to less conventional domains and problems. We describe our experimental conditions in Section 3. We compare the performance of token and segment based transduction grammar models in Section 4. We compare our data selection schemes and disfluency handling strategies in Sections 5 and 6. Finally, in

[1] Also known in SMT as "synchronous grammars".

[2] Particularly the repetitive chants, exclamations, and interjections in hip hop "hype man" style backing vocals.

Section 7 we describe some preliminary results obtained using our approach on improvising hip hop responses in French and conclude in Section 8.

2 Related work

Although a few attempts have been made to apply statistical NLP learning methods to unconventional domains, Freestyle is among the first to tackle the genre of hip hop lyrics (Addanki and Wu, 2013; Wu *et al.*, 2013a,b). Our preliminary work suggested the need for further research to identify models that capture the correct generalizations to be able to generate fluent and rhyming responses. As a step towards this direction, we contrast the performance of interpolated bottom-up token based rule induction and top-down segmenting transduction grammar models and token based transduction grammar models. We briefly describe some of the past work in statistical NLP on unconventional domains below.

Most of the past work either uses some form of prior linguistic knowledge or enforces harsher constraints such as set number of words in a line, or a set meter which are warranted by more structured domains such as poetry. However, in hip hop lyrics it is hard to make any linguistic or structural assumptions. For example, words such as sho, flo, holla which frequently appear in the lyrics are not part of any standard lexicon and hip hop does not require a set number of syllables in a line, unlike poems. Also, surprising and unlikely rhymes in hip hop are frequently achieved via intonation and assonance, making it hard to apply prior phonological constraints.

A phrase based SMT system was trained to "translate" the first line of a Chinese couplet or *duilian* into the second by Jiang and Zhou (2008). The most suitable next line was selected by applying linguistic constraints to the n best output of the SMT system. However in contrast to Chinese couplets, which adhere to strict rules requiring, for example, an identical number of characters in each line and one-to-one correspondence in their metrical length, the domain of hip hop lyrics is far more unstructured and there exists no clear constraint that would ensure fluent and rhyming responses to hip hop challenge lyrics. Barbieri *et al.* (2012) use controlled Markov processes to semi-automatically generate lyrics that satisfy the structural constraints of rhyme and meter. Tamil lyrics were automatically generated given a melody using conditional random fields by A. *et al.* (2009). The lyrics were represented as a sequence of labels using the *KNM* system where *K*, *N* and *M* represented the long vowels, short vowels and consonants respectively.

Genzel *et al.* (2010) used SMT in conjunction with stress patterns and rhymes found in a pronunciation dictionary to produce translations of poems. Although many constraints were applied in translating full verses of poems, it was challenging to satisfy all the constraints. Stress patterns were assigned to words given the meter of a line in Shakespeare's sonnets by Greene *et al.* (2010), which were then combined with a language model to generate poems. Sonderegger (2011) attempted to infer the pronunciation of words in old English by identifying the rhyming patterns using graph theory. However, their heuristic of clustering words with similar IPA endings resulted in large clusters of false positives such as bloom and numb. A language-independent generative model for stanzas in poetry was proposed by Reddy and Knight (2011) via which they could discover rhyme schemes in French and English poetry.

3 Experimental conditions

Before introducing our Freestyle models, we first detail our experimental assumptions and the evaluation scheme under which the responses generated by different models are compared against one another. We describe our training data as well as a phrase-based SMT (PBSMT) contrastive baseline. We also define the evaluation scheme used to compare the responses of different systems on criteria of fluency and rhyming.

3.1 Training data

We used freely available user generated hip hop lyrics on the Internet to provide training data for our experiments. We collected approximately 52,000 English hip hop song lyrics amounting to approximately 800Mb of raw HTML content. The data was cleaned by stripping HTML tags, metadata and normalized for special characters and case differences. The processed corpus contained 22 million tokens with 260,000 verses and 2.7 million lines of hip hop lyrics. 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.

3.2 Evaluation scheme

The performance of various FREESTYLE versions was evaluated on the task of generating a improvised fluent and rhyming response given a single line of a hip hop verse as a challenge. The output of all the systems on the test set was given to three independent frequent hip hop listeners for manual evaluation. They were asked to evaluate the system outputs according to fluency and the degree of rhyming. They were free to choose the tune to make the lyrics rhyme as the beats of the song were not used in the training data. Each evaluator was asked to score the response of each system on the criterion of fluency and rhyming as being *good*, *acceptable* or *bad*.

3.3 Phrase-based SMT baseline

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 (Koehn *et al.*, 2007) was also trained in order to compare its performance with our transduction grammar induction model. A 4-gram language model which was trained on the entire training corpus using SRILM (Stolcke, 2002) was used to generate responses in conjunction with the phrase-based translation model. As no automatic quality evaluation metrics exist for hip hop responses analogous to BLEU for SMT, the model weights cannot be tuned in conventional ways such as MERT (Och, 2003). Instead, a slightly higher than typical language model weight was empirically chosen using a small development set to produce fluent outputs.

4 Interpolated segmenting model vs. token based model

We compare the performance of transduction grammars induced via interpolated token based and rule segmenting (ISTG) versus token based transduction grammars (TG) on the task of generating a rhyming and fluent response to hip hop challenges. We use the framework of stochastic transduction grammars, specifically bracketing ITGs (inversion transduction grammars) (Wu, 1997), as our translation model for "transducing" any given challenge into a rhyming and fluent response. Our choice is motivated by the significant amount of empirical evidence for the representational capacity of transduction grammars across a spectrum of natural language tasks such as textual entailment (Wu, 2006), mining parallel sentences (Wu and Fung, 2005) and machine translation (Zens and Ney, 2003). Further, existence of efficient learning algorithms (Saers *et al.*, 2012; Saers and Wu, 2011) that make no language specific assumptions, make inversion transduction grammars a suitable framework for our modeling needs. Examples of lexical transduction rules can be seen in Tables 3 and 5. In addition, the grammar also includes structural transduction rules for the straight case $A \rightarrow [A\ A]$ and also the inverted case $A \rightarrow \langle A\ A \rangle$.

4.1 Token based vs. segmental ITGs

The degenerate case of ITGs are token based ITGs wherein each translation rule contains at most one token in input and output languages. Efficient induction algorithms with polynomial run time exist for token based ITGs and the expressivity they permit has been empirically determined to capture most of the word alignments that occur across natural languages. The parameters of the token based ITGs can be estimated using expectation maximization through an efficient dynamic programming algorithm in conjunction with beam pruning (Saers and Wu, 2011).

In contrast to token based ITGs, each rule in a segmental ITG grammar can contain more than one token in both input and output languages. In machine translation applications, segmental models produce translations that are more fluent as they can capture lexical knowledge at a phrasal level. However, only a handful of purely unsupervised algorithms exist for learning segmental ITGs under matched training and testing assumptions. Most other approaches in SMT use a variety of ad hoc heuristics for extracting segments from token alignments, justified purely by short term improvements in automatic MT evaluation metrics such as BLEU (Papineni *et al.*, 2002) which cannot be transferred to our current task. Instead, we use a completely unsupervised learning algorithm for segmental ITGs that stays strictly within the transduction grammar optimization framework for both training and testing as proposed in Saers

et al. (2013).

Saers *et al.* (2013) induce a phrasal inversion transduction grammar via interpolating the bottom-up rule chunking approach proposed in Saers *et al.* (2012) with a top-down rule segmenting approach driven by a minimum description length objective function (Solomonoff, 1959; Rissanen, 1983) that trades off the maximum likelihood against model size. Saers *et al.* (2013) report improvements in BLEU score (Papineni *et al.*, 2002) on their translation task. In our current approach instead of using a bottom-up rule chunking approach we use a simpler token based grammar instead. Given two grammars (G_a and G_b) and an interpolation parameter α the probability function of the interpolated grammar is given by:

$$p_{a+b}(r) = \alpha p_a(r) + (1-\alpha) p_b(r)$$

for all rules r in the union of the two rule sets, and where p_{a+b} is the rule probability function of the combined grammar and p_a and p_b are the rule probability functions of G_a and G_b respectively. The pseudocode for the top-down rule segmenting algorithm is shown in 1. The algorithm uses the methods `collect_biaffixes`, `eval_dl`, `sort_by_delta` and `make_segmentations`. These methods collect all the biaffixes in an ITG, evaluate the difference in description length, sort candidates by these differences, and commit to a given set of candidates, respectively. The suitable interpolation parameter is chosen empirically based on the responses generated on a small development set.

We compare the performance of inducing a token based ITG versus inducing a segmental ITG using interpolated bottom-up token based rule induction and top-down rule segmentation. To highlight some of the inherent challenges in adapting other algorithms to this novel task, we also compare the quality of the responses generated by our model to those generated by an off-the-shelf phrase based SMT system.

Algorithm 1 Iterative rule segmenting learning driven by minimum description length.

```
Φ                                          ▷ The ITG being induced
repeat
    δ_sum ← 0
    bs ← collect_biaffixes(Φ)
    bδ ← []
    for all b ∈ bs do
        δ ← eval_dl(b, Φ)
        if δ < 0 then
            bδ ← [bδ, ⟨b, δ⟩]
    sort_by_delta(bδ)
    for all ⟨b, δ⟩ ∈ bδ do
        δ′ ← eval_dl(b, Φ)
        if δ′ < 0 then
            Φ ← make_segmentations(b, Φ)
            δ_sum ← δ_sum + δ′
until δ_sum ≥ 0
return Φ
```

4.2 Decoding heuristics

We use our in-house ITG decoder implemented according to the algorithm mentioned in Wu (1996) for the generating responses to challenges by decoding with the trained transduction grammars. The decoder uses a CKY-style parsing algorithm (Cocke, 1969) with cube pruning (Chiang, 2007). The decoder builds an efficient hypergraph structure which is then scored using the induced grammar. The trained transduction grammar model was decoded using the 4-gram language model and the model weights determined as described in 3.3.

In our decoding algorithm, we restrict the reordering to only be monotonic as we want to produce output that follows the same rhyming order of the challenge. Interleaved rhyming order is harder to evaluate without the larger context of the song and we do not address that problem in our current model. We also penalize singleton rules to produce responses of similar length as successive lines in a stanza are typically of similar length. Finally, we add a penalty to *reflexive* translation rules that map the same surface form to itself such as $A \rightarrow$ yo/yo. We obtain these rules with a high probability due to the presence of sentence pairs where both the input and output are identical strings as many stanzas in our data contain repeated chorus lines.

4.3 Results: Rule segmentation improves responses

Results in Table 1 indicate that the ISTG outperforms the TG model towards the task of generating fluent and rhyming responses. On the criterion of fluency,

Table 1: Percentage of ≥*good* and ≥*acceptable* (i.e., either good or acceptable) responses on fluency and rhyming criteria. PBSMT, TG and ISTG models trained using corpus generated from all adjacent lines in a verse. PBSMT+RS, TG+RS, ISTG+RS are models trained on rhyme scheme based corpus selection strategy. Disfluency correction strategy was used in all cases.

model	*fluency (≥good)*	*fluency (≥acceptable)*	*rhyming (≥good)*	*rhyming (≥acceptable)*
PBSMT	3.14%	4.70%	1.57%	4.31%
TG	21.18%	54.51%	23.53%	39.21%
ISTG	**26.27%**	**57.64%**	**27.45%**	**48.23%**
PBSMT+RS	30.59%	43.53%	1.96%	9.02%
TG+RS	**34.12%**	60.39%	20.00%	42.74%
ISTG+RS	30.98%	**61.18%**	**30.98%**	**53.72%**

Table 2: Transduction rules learned by ISTG model.

transduction grammar rule	*log prob.*
$A \rightarrow$ long/wrong	-11.6747
$A \rightarrow$ rhyme/time	-11.6604
$A \rightarrow$ felt bad/couldn't see what i really had	-11.3196
$A \rightarrow$ matter what you say/leaving anyway	-11.8792
$A \rightarrow$ arhythamatic/this rhythm is sick	-12.3492

the ISTG model produces a significantly higher fraction of sentences rated *good* (26.27% vs. 21.18%) and ≥*acceptable* (57.64% vs. 54.51%). Higher fraction of responses generated by the ISTG model are rated as *good* (27.45% vs. 23.53%) and ≥*acceptable* (57.64% vs. 54.51%) compared to the TG model. Both TG and ISTG model perform significantly better than the PBSMT baseline. Upon inspecting the learned rules, we noticed that the ISTG models capture rhyming correspondences both at the token and segmental levels. Table 2 shows some examples of the transduction rules learned by ISTG grammar trained using rhyme scheme detection.

5 Data selection via rhyme scheme detection vs. adjacent lines

We now compare two data selection approaches for generating the training data for transduction grammar induction via a rhyme scheme detection module and choosing all adjacent lines in a verse. We also briefly describe the training of the rhyme scheme detection module and determine the efficacy of our data selection scheme by training the ISTG model, TG model and the PBSMT baseline on training data generated with and without employing the rhyme scheme detection module. As the rule segmenting approach was intended to improve the fluency as opposed to the rhyming nature of the responses, we only train the rule segmenting model on the randomly chosen subset of all adjacent lines in the verse. Further, adding adjacent lines as the training data to the segmenting model maintains the context of the responses generated thereby producing higher quality responses. The segmental transduction grammar model was combined with the token based transduction grammar model trained on data selected with and without using rhyme scheme detection model.

5.1 Rhyme scheme detection

Although our approach adapts a transduction grammar induction model toward the problem of generating fluent and rhyming hip hop responses, it would be undesirable to train the model directly on all the successive lines of the verses—as done by Jiang and Zhou (2008)—due to variance in hip hop rhyming patterns. For example, adding successive lines of a stanza which follows **ABAB** rhyme scheme as training instances to the transduction grammar causes incorrect rhyme correspondences to be learned. The fact that a verse (which is usually represented as a separate paragraph) may contain multiple stanzas of varying length and rhyme schemes worsens this problem. Adding all possible pairs of lines in a verse as training examples not only introduces a lot of noise but also explodes the size of the training data due to the large size of the verse.

We employ a rhyme scheme detection model (Addanki and Wu, 2013) in order to select training instances that are likely to rhyme. Lines belonging to

the same stanza and marked as rhyming according to the rhyme scheme detection model are added to the training corpus. We believe that this data selection scheme will improve the rhyming associations learned during the transduction grammar induction thereby biasing the model towards producing fluent and rhyming output.

The rhyme scheme detection model proposes a HMM based generative model for a verse of hip hop lyrics similar to Reddy and Knight (2011). However, owing to the lack of well-defined verse structure in hip hop, a number of hidden states corresponding to stanzas of varying length are used to automatically obtain a *soft-segmentation* of the verse. Each state in the HMM corresponds to a stanza with a particular rhyme scheme such as **AA**, **ABAB**, **AAAA** while the emissions correspond to the final words in the stanza. We restrict the maximum length of a stanza to be four to maintain a tractable number of states and further only use states to represent stanzas whose rhyme schemes could not be partitioned into smaller schemes without losing a rhyme correspondence.

The parameters of the HMM are estimated using the EM algorithm (Devijer, 1985) using the corpus generated by taking the final word of each line in the hip hop lyrics. The lines from each stanza that rhyme with each other according to the Viterbi parse using the trained model are added as training instances for transduction grammar induction. As the source and target languages are identical, each selected pair generates two training instances: a challenge-response and a response-challenge pair.

The training data for the rhyme scheme detector was obtained by extracting the end-of-line tokens from each verse. However, upon data inspection we noticed that shorter lines in hip hop stanzas are typically joined with a comma and represented as a single line of text and hence all the tokens before the commas were also added to the training corpus. We obtained a corpus containing 4.2 million tokens corresponding to potential rhyming candidates comprising of around 153,000 unique token types.

We evaluated the performance of our rhyme scheme detector on the task of correctly labeling a given verse with rhyme schemes. As our model is completely unsupervised, we chose a random sample of 75 verses from our training data as our test set. Two native English speakers who were frequent hip hop listeners were asked to partition the verse into stanzas and assign them with a gold standard rhyme scheme. Precision and recall were aggregated for the Viterbi parse of each verse against this gold standard and f-score was calculated. The rhyme scheme detection module employed in our data selection obtained a precision of 35.81% and a recall of 57.25%, giving an f-score of 44.06%.

5.2 Training data selection via rhyme scheme detection

We obtained around 600,000 training instances upon extracting a training corpus using rhyme scheme detection module as described in Section 5.1. We added those lines that were adjacent *and* labeled as rhyming by the rhyme scheme detector as training instances resulting in a training corpus of size 200,000.

5.3 Training data selection via adjacent lines

Considering all adjacent lines in a verse resulted in a corpus with over 5 million training instances. In order to ensure fair comparison of models trained on data selected using rhyme scheme detection, we randomly chose 200,000 training instances from the generated corpus. The training corpus thus generated shared around 15% of training instances with the corpus generated through our proposed data selection scheme.

5.4 Results: Rhyme scheme detection helps

Results in Table 1 indicate that using the rhyme scheme detector for training data selection helps produce significantly more fluent responses compared to using adjacent lines. A possible explanation for this could be that adding all adjacent lines as training instances introduces a lot of noise into the model which hurts the fluency of the responses generated. Also, the cumulative fraction of sentences that were labeled *good* or ≥*acceptable* on the criterion of rhyming is larger when rhyme scheme detection was used to generate the training data (although the TG model trained on the corpus generated using adjacent lines produces a higher percentage of rhyming responses that were rated *good*). Given the significantly higher rate of response fluency when using rhyme scheme detection, we argue that using rhyme scheme detector for data selection is beneficial.

Table 3: English hip hop challenge-response examples.

challenge	man i **die** to **see** em all thun i just don't **care**
TG+RS	in the **sky** and **me** the in polla and the you **there**
TG	and the ride the me the and white the i the **air**
PBSMT+RS	man i live to see em all i just don't care
challenge	did a twelve year **bid** in the streets and held it **down**
TG+RS	to the girls here **kid** the and to the thought the **now**
TG	to the p's here **did** the a the i was the the
PBSMT+RS	did a year in the streets and it down
challenge	oh i believe in **yesterday**
TG+RS	can you see the **day**
TG	now you see the **way**
PBSMT+RS	oh i believe in tomorrow
challenge	what would i **do**
TG+RS	just me and **you**
TG	and you and **you**
PBSMT+RS	what would you do
challenge	cause you ain't going **home** till the early **morn**
TG+RS	and the you this **alone** i i gotta **on**
TG	and i you my on the a home we
PBSMT+RS	cause you and your friends aint nothing but

It is also interesting to note from Table 1 that ISTG+RS performs better than TG+RS indicating that transduction grammar induced via interpolating token based grammar and rule segmenting produces better responses than token based transduction grammar on both data selection schemes. Although the average fraction of responses rated *good* on fluency are slightly lower for ISTG+RS compared to TG+RS (34.12% vs. 30.98%), the fraction of responses rated ≥*acceptable* are higher (61.18% vs. 57.64%). It is important to note that the fraction of sentences rated *good* and ≥*acceptable* on rhyming are much larger for ISTG+RS model. Although the fluency of the responses generated by PBSMT+RS drastically improves compared to PBSMT it still lags behind the TG+RS and ISTG+RS models on both fluency and rhyming. The results in Table 1 confirm our hypothesis that off-the-shelf SMT systems are not guaranteed to be effective on our novel task.

5.5 Challenge-response examples

Table 3 shows some of the challenges and the corresponding responses of PBSMT+RS, TG+RS and TG model. While PBSMT+RS and TG+RS models generate responses reflecting a high degree of fluency, the output of the TG contains a lot of articles. It is interesting to note that TG+RS produces responses comparable to PBSMT+RS despite being a token based transduction grammar. However, PBSMT tends to produce responses that are too similar to the challenge. Moreover, TG models produce responses that indeed rhyme better (shown in boldface). In fact, TG tries to rhyme words not only at the end but also in middle of the lines, as our transduction grammar model captures structural associations more effectively than the phrase-based model.

6 Disfluency handling via disfluency correction and filtering

In this section, we compare the effect of two disfluency mitigating strategies on the quality of the responses generated by the PBSMT baseline and token based transduction grammar model with and without using rhyme scheme detection.

6.1 Correction vs. filtering

Error analysis of our initial runs showed a disturbingly high proportion of responses generated by our system that contained disfluencies with successive repetitions of words such as the and I. Upon inspection of data we noticed that the training lyrics actually did contain such disfluencies and backing vocal lines, amounting to 10% of our training data. We therefore compared two alternative strategies to tackle this problem. The first strategy involved filtering out all lines from our training corpus which contained such disfluencies. In the second strategy, we implemented a disfluency detection and correction algorithm (for example, the the the, which frequently occurred in the training corpus, was corrected to simply the). The PBSMT baseline and the TG model were trained on both the filtered and corrected versions of the training corpus and the quality of the responses were compared.

6.2 Results: Disfluency correction helps

The results in Table 4 indicate that the disfluency correction strategy outperforms the filtering strategy for both TG and TG+RS models. For the model TG+RS, disfluency correction generated 34.12% *good* responses in terms of fluency, while the filtering strategy produced only 28.63% *good* responses. Similarly for the model TG, disfluency correction produced 21.8% of responses with *good* fluency and the filtering strategy produced only 17.25%. Disfluency correction strategy produces higher fraction of responses with ≥*acceptable* fluency compared to the filtering strategy for both TG and TG+RS models. This result is not surprising, as harshly pruning

Table 4: Effect of the disfluency correction strategies on fluency of the responses generated for the TG induction models vs PBSMT baselines using both rhyme scheme detection and adjacent lines as the corpus selection method.

model+disfluency strat.	fluency (good)	fluency (≥acceptable)	rhyming (good)	rhyming (≥acceptable)
PBSMT+filtering	**4.3%**	**13.72%**	3.53%	7.06%
PBSMT+correction	3.14%	4.70%	1.57%	4.31%
PBSMT+RS+filtering	**31.76%**	**43.91%**	12.15%	21.17%
PBSMT+RS+correction	30.59%	43.53%	1.96%	9.02%
TG+filtering	17.25%	46.27%	18.04%	33.33%
TG+correction	**21.18%**	**54.51%**	23.53%	39.21%
TG+RS+filtering	28.63%	56.86%	14.90%	34.51%
TG+RS+correction	**34.12%**	**60.39%**	20.00%	42.74%

the training corpus causes useful word association information necessary for rhyming to be lost. Surprisingly, for both PBSMT and PBSMT+RS models, the disfluency correction has a negative effect on the fluency level of the response but still falls behind TG and TG+RS models. As disfluency correction yields more fluent responses for TG and TG+RS models, the results for ISTG and ISTG+RS models in Table 1 were obtained using disfluency correction strategy.

7 Maghrebi French hip hop

We have begun to apply Freestyle to rap in languages other than English, taking advantage of the language independence and linguistics-light approach of our unsupervised transduction grammar induction methods. With no special adaption our transduction grammar based model performs surprisingly well, even with significantly smaller training data size and noisier data. These results across different languages are encouraging as they can be used to discover truly language independence assumptions. We briefly describe our initial experiments on Maghrebi French hip hop lyrics below.

7.1 Dataset

We collected freely available French hip hop lyrics of approximately 1300 songs. About 85% of the songs were by Maghrebi French artists of Algerian, Moroccan, or Tunisian cultural backgrounds, while the remaining were by artists from the rest of Francophonie. As the large majority of songs are in Maghrebi French, the lyrics are sometimes interspersed with romanized Arabic such as "De la traversée du désert au bon couscous de **Yéma**" (**Yéma** means *My mother*). Some songs also contain Berber phrases, for instance "**a yemmi ino, a thizizwith**" (which means *my son, a bee*). Furthermore, some songs also contained English phrases in the style of gangster rap such as "T'es **game over**, **game over**... Le son de **Chicken wings**". As mentioned earlier, it is complexity like this which dissuaded us from making language specific assumptions in our model.

We extracted the end-of-line words and obtained a corpus containing 120,000 tokens corresponding to potential rhyming candidates with around 29,000 unique token types which was used as the training data for the rhyme scheme detector module. For the transduction grammar induction, the training data contained about 47,000 sentence pairs selected using rhyme scheme detection.

7.2 Results

After human evaluation by native French speakers and frequent hip hop listeners, our transduction grammar based model generates about 9.2% and 14.5% of the responses that are rated *good* by the human evaluators on the criterion of fluency and rhyming respectively. About 30.2% and 38% of the responses are rated as ≥*acceptable*. These numbers are encouraging given the noisy lyrics and much smaller amount of training data. Some examples of the challenge-response pairs and learned transduction rules in French are shown in Tables 5 and 6.

From Table 5, we can see that responses generated by the system rhyme with the challenges. The first response is fluent and not only perfectly rhymes with the challenge but also semantically valid. In the second example, the model realizes a less common **AABA** rhyme scheme through the response. The re-

Table 5: French hip hop challenge-response examples.

challenge	Si je me **trompe**
response	faut que je **raconte**
challenge	Un jour je suis un **livre**
response	et ce que je de **vivre**
challenge	Pacha mama ils ne voient pas ta **souffrance**
response	Combat ni leur de voulait de la **décadence**
challenge	la palestine n'etait pas une **terre** sans peuple.
response	le **darfour d'autre** de la **guerre** on est
challenge	Une **banlieue** qui **meut**
response	les **yeux** et

Table 6: Transduction rules for Maghrebi French hip hop.

transduction grammar rule	*log prob.*
$A \rightarrow$ terre/la guerre	-9.4837
$A \rightarrow$ haine/peine	-9.77056
$A \rightarrow$ mal/pays natal	-10.6877
$A \rightarrow$ je frissonne/mi corazon	-11.0931
$A \rightarrow$ gratteurs/rappeurs	-11.7306

sponse in the third example, exhibits strong rhyming with the challenge and both the challenge and the response contain words like souffrance, combat and décadence which are related. Similarly in the fourth example, the challenge and response also contain semantically related tokens which also rhyme. These examples illustrate that our transduction grammar formalism coupled with our rhyme scheme detection module does capture the necessary correspondences between lines of hip hop lyrics without assuming any language specific resources.

8 Conclusion

We presented a new machine learning approach for improvising hip hop responses to challenge lyrics by inducing stochastic transduction grammars, and demonstrated that inducing the transduction rules by interpolating bottom-up token based rule induction and rule segmentation strategies outperforms a token based baseline. We compared the performance of our Freestyle model against a widely used off-the-shelf phrase-based SMT model, showing that PB-SMT falls short in tackling the noisy and highly unstructured domain of hip hop lyrics. We showed that the quality of responses improves when the training data for the transduction grammar induction is selected using a rhyme scheme detector. Several domain related oddities such as disfluencies and backing vocals have been identified and some strategies for alleviating their effects have been compared. We also reported results on Maghrebi French hip hop lyrics which indicate that our model works surprisingly well with no special adaptation for languages other than English. In the future, we plan to investigate alternative training data selection techniques, disfluency handling strategies, search heuristics, and novel transduction grammar induction models.

Acknowledgements

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

References

Ananth Ramakrishnan A., Sankar Kuppan, and Lalitha Devi Sobha. "Automatic generation of Tamil lyrics for melodies." *Workshop on Computational Approaches to Linguistic Creativity (CALC-09)*. 2009.

Karteek Addanki and Dekai Wu. "Unsupervised rhyme scheme identification in hip hop lyrics using hidden Markov models." *1st International Conference on Statistical Language and Speech Processing (SLSP 2013)*. 2013.

Gabriele Barbieri, François Pachet, Pierre Roy, and Mirko Degli Esposti. "Markov constraints for generating lyrics with style." *20th European Conference on Artificial Intelligence, (ECAI 2012)*. 2012.

David Chiang. "Hierarchical phrase-based translation." *Computational Linguistics*, 33(2), 2007.

John Cocke. *Programming languages and their compilers: Preliminary notes*. Courant Institute of Mathematical Sciences, New York University, 1969.

P.A. Devijer. "Baum's forward-backward algorithm revisited." *Pattern Recognition Letters*, 3(6), 1985.

D. Genzel, J. Uszkoreit, and F. Och. "Poetic statistical machine translation: rhyme and meter." *2010 Conference on Empirical Methods in Natural Language Processing (EMNLP 2010)*. Association for Computational Linguistics, 2010.

E. Greene, T. Bodrumlu, and K. Knight. "Automatic analysis of rhythmic poetry with applications

to generation and translation." *2010 Conference on Empirical Methods in Natural Language Processing (EMNLP 2010)*. Association for Computational Linguistics, 2010.

Long Jiang and Ming Zhou. "Generating Chinese couplets using a statistical MT approach." *22nd International Conference on Computational Linguistics (COLING 2008)*. 2008.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. "Moses: Open source toolkit for statistical machine translation." *Interactive Poster and Demonstration Sessions of the 45th Annual Meeting of the Association for Computational Linguistics (ACL 2007)*. June 2007.

Mark Liberman. "Rap scholarship, rap meter, and the anthology of mondegreens." `http://languagelog.ldc.upenn.edu/nll/?p=2824`, December 2010. Accessed: 2013-06-30.

Franz Josef Och. "Minimum error rate training in statistical machine translation." *41st Annual Meeting of the Association for Computational Linguistics (ACL-2003)*. July 2003.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. "BLEU: a method for automatic evaluation of machine translation." *40th Annual Meeting of the Association for Computational Linguistics (ACL-02)*. July 2002.

S. Reddy and K. Knight. "Unsupervised discovery of rhyme schemes." *49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL HLT 2011)*, vol. 2. Association for Computational Linguistics, 2011.

Jorma Rissanen. "A universal prior for integers and estimation by minimum description length." *The Annals of Statistics*, 11(2), June 1983.

Markus Saers, Karteek Addanki, and Dekai Wu. "From finite-state to inversion transductions: Toward unsupervised bilingual grammar induction." *24th International Conference on Computational Linguistics (COLING 2012)*. December 2012.

Markus Saers, Karteek Addanki, and Dekai Wu. "Combining top-down and bottom-up search for unsupervised induction of transduction grammars." *Seventh Workshop on Syntax, Semantics and Structure in Statistical Translation (SSST-7)*. June 2013.

Markus Saers and Dekai Wu. "Reestimation of reified rules in semiring parsing and biparsing." *Fifth Workshop on Syntax, Semantics and Structure in Statistical Translation (SSST-5)*. Association for Computational Linguistics, June 2011.

Ray J. Solomonoff. "A new method for discovering the grammars of phrase structure languages." *International Federation for Information Processing Congress (IFIP)*. 1959.

M. Sonderegger. "Applications of graph theory to an English rhyming corpus." *Computer Speech & Language*, 25(3), 2011.

Andreas Stolcke. "SRILM – an extensible language modeling toolkit." *7th International Conference on Spoken Language Processing (ICSLP2002 - INTERSPEECH 2002)*. September 2002.

Dekai Wu. "A polynomial-time algorithm for statistical machine translation." *34th Annual Meeting of the Association for Computational Linguistics (ACL96)*. 1996.

Dekai Wu. "Stochastic inversion transduction grammars and bilingual parsing of parallel corpora." *Computational Linguistics*, 23(3), 1997.

Dekai Wu. "Textual entailment recognition using inversion transduction grammars." Joaquin Quiñonero-Candela, Ido Dagan, Bernardo Magnini, and Florence d'Alché Buc (eds.), *Machine Learning Challenges, Evaluating Predictive Uncertainty, Visual Object Classification and Recognizing Textual Entailment, First PASCAL Machine Learning Challenges Workshop (MLCW 2005)*, vol. 3944 of *Lecture Notes in Computer Science*. Springer, 2006.

Dekai Wu, Karteek Addanki, and Markus Saers. "FREESTYLE: A challenge-response system for hip hop lyrics via unsupervised induction of stochastic transduction grammars." *14th Annual Conference of the International Speech Communication Association (Interspeech 2013)*. 2013a.

Dekai Wu, Karteek Addanki, and Markus Saers. "Modeling hip hop challenge-response lyrics as machine translation." *14th Machine Translation Summit (MT Summit XIV)*. 2013b.

Dekai Wu and Pascale Fung. "Inversion transduction grammar constraints for mining parallel sentences from quasi-comparable corpora." *Second International Joint Conference on Natural Language Processing (IJCNLP 2005)*. Springer, 2005.

Richard Zens and Hermann Ney. "A comparative study on reordering constraints in statistical machine translation." *41st Annual Meeting of the Association for Computational Linguistics (ACL-2003)*. Association for Computational Linguistics, 2003.