Chapter 6

Poetry generation

Using the rhyme scheme annotated data set described in chapter 3, and the poetry generation model described in chapter 5, we train an LSTM-based poetry generation model.

6.1 Model architecture

Our model is loosely based on Lau et al. (2018). We also use LSTMs for both text generation and rhyme modelling. Similarly to Lau et al. (2018), we sample rhyming line ending words that correspond to a rhyme scheme, and generate the rest of the stanza based on those words. However, we do not explicitly model the stress or pentimeter in the source poetry. The only poetry feature that we explicitly model is rhyme, the rest we trust the language model to capture, if it is able to do so. And while Lau et al. (2018) is strictly focused on Shakespearean sonnets, with a strict rhyme scheme, our model can generate a stanza from any rhyme scheme.

The input for the poetry generation model is a rhyme scheme, and the output is a stanza with a rhyme pattern that matches the rhyme scheme. The model consists of a language model and a rhyme model, and the stanza is generated using both. We create two slightly different versions of the poetry generation model, depending on if the language model is trained on whole stanzas or on lines from stanzas. The rhyme model is the same for both models, which is the rhyme generation model trained on the dense buckets of size $>= 10$ (as described in Section 5.2.2 in the previous chapter).

To explain the poetry generation model, we first describe the language model:

6.1.1 Language model

Our language model is an LSTM-based language model for natural language generation. The model architecture is a sequential model with an embedding layer, an LSTM layer, and an output layer the size of the vocabulary (see Figure 6.1). For each time step during generation, the input is the word embedding of a word, and the output is the probability of each word in the vocab being the next word in the sequence.

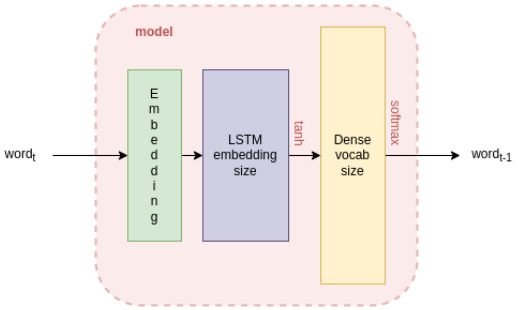

Figure 6.1: Language model architecture

X	y
<s>	jeg
<s>jeg	liker
<s>jeg liker	fisk
<s>jeg liker fisk	</s>

Table 6.1: Teacher forcing training data for the sentence "jeg liker fisk"

The model is trained using *teacher forcing*; each instance of training data is a sequence of words, and the target is the next word in the sequence. Thus, for the sentence "jeg liker fisk", with teacher forcing the training data would look like in Table 6.1. '<s>' and '</s>' are special tokens to mark sequence start and sequence end, respectively.

For a model trained with teacher forcing like this, simply inputting the sequence start token starts generation. If the next predicted token is the sequence end token, generation stops and the full generated sequence is returned.

As we already mentioned, we train two different language models, a line based one and a stanza based one. For the line based model, each stanza is split on line breaks, and training data is constructed for each line. That is, the sequence start and sequence end markers are placed on either side of each line, and training data is constructed as in Table 6.1.

For the stanza based model, the start and end of sequence tokens are placed on either size of each stanza. Here we introduce another special token, '<n>' representing the line break. The idea is that this way, the model will produce stanzas with a consistent theme throughout the stanza. In contrast to this, the line based model can only access the context of each line. To create a stanza with the line based model, one has to generate the number of lines wanted in the stanza separately, which can lead to the content of the lines being totally unrelated to each other.

An important detail is that our models are trained on reversed data. Instead of predicting the next word, they predict the previous word. This way, we can enforce rhyme by first deciding rhyming line ending words, and generate the rest of the sequences backwards.

```
on input rhyme_scheme:
    for each unique symbol in rhyme_scheme:
        language_model.generate_line_ending_word().

    for each line ending word w:
        rhyme_model.get_bucket(w)

    for each symbol in rhyme_scheme:
        sample a line_ending word from the correct bucket

    # Now there is one line-ending-word for each line, and
    # rhyme relations are according to rhyme_scheme

    for each line ending word w:
        generate_line(w)
        (and reverse line)

    return the lines
```

Listing 3: Simplified pseudocode for line based stanza generation

Language model training

We reason that for this task, overfitting on the training data is not a problem, as we want the model to learn the source corpus and produce similar texts. This is not a model that needs to be able to generalize and be applied to unseen texts. We therefore train both language models on the entire corpus data, without splitting it into training and test sets. Both models are trained for 100 epochs, with a batch size of 256. The optimizer is adam, and the loss function sparse categorical crossentropy.

6.1.2 Generating rhyming stanzas

Now that we have described how the language models work, we can explain the full poetry generation model. Recall from chapter 5 that the rhyme generation model returns a *bucket* of rhyming words on any word input.

For the line based poetry generation model, the basic stanza generation algorithm works as seen in Listing 3. The poetry generation model takes in a rhyme scheme, and using the rhyme scheme, generates the appropriate number of rhyming line-ending words using the language and rhyme generation models. Then, each line is generated backwards, the prompt being the end of sequence marker and the line-ending word. The generated lines are stacked together and the result is a stanza following the given rhyme scheme.

The stanza based approach is a little bit more complicated. Here the method for sampling from the language model is modified so that the sequence is returned when the line break token is predicted. The next line ending word is inserted into the sequence, and it is sent back to the language model to continue

```
on input rhyme_scheme:
    for each unique symbol in rhyme_scheme:
        language_model.generate_line_ending_word().

    for each line ending word w:
        rhyme_model.get_bucket(w)

    for each symbol in rhyme_scheme:
        sample a line_ending word from the correct bucket

    # Now there is one line-ending-word for each line, and
    # rhyme relations are according to rhyme_scheme

    stanza_so_far = ""
    for each line ending word w:
        stanza_so_far += w
        g = language_model.generate_line(stanza_so_far)
        stanza_so_far += g

    return stanza_so_far
```

Listing 4: Simplified pseudocode for stanza based stanza generation

generation. This way, the rhyme scheme is ensured, and the context of the whole stanza (so far) is available to the language generation model. See listing 4 for pseudocode for the generation function using the stanza based language model. The full code for both models can be found on the master thesis github[1]

A parameter to the poetry generation algorithm that is not shown in the simplified algorithms above is the temperature-parameter. In the language model, the default is to choose the most likely next word for a given input word during generation. With temperature, the output probabilities are changed slightly using a multinomial probability distribution. This ensures higher exploration, as the choice of the next word is no longer completely deterministic. The higher the temperature, the more exploration. For our poetry generation models, the stanzas are produced using a temperature of 0.5.

[1] https://github.com/titaenstad/mester

6.2 Evaluation

As stated in the introduction, our modelling objectives are to find out if we can create a Norwegian poetry generation model that produces poetry of a quality such that

1. it can't be discerned from poetry written by humans and
2. it rhymes.

6.2.1 Approach

As in the earlier works explored in chapter 2, we also make use of human evaluation in order to evaluate the generated poetry. We use both side-by-side and standalone evaluation. In side-by-side evaluation, the evaluator is presented with two stanzas, and asked to pick out the one that is written by a human. In standalone evaluation, the evaluator is presented with one stanza at the time, and is asked whether or not they believe it is written by a human.

They are also asked to rate how well each stanza rhymes, on a scale of 0 to 3. 0 means that the stanza contains no rhyme. 1 means that the stanza contains some almost-rhyme. 2 means that the stanza has some rhyme, for example that two lines of the stanza rhyme, but the rest do not. 3 is used when is it obvious that the stanza was written to rhyme. See Appendix **??** for the full evaluation instructions.

We create 4 forms, to minimize the work load of the evaluation, and this way get more people to participate. We generate 40 stanzas with each model, 10 stanzas for each form. We use the top 10 most frequent rhyme schemes, and generate 4 stanzas for each scheme. The models are the baseline model (the stanza based language model with no rhyme component), the line-level poetry generation model and the stanza-level poetry generation model. 5 stanzas from each model are used for side-by-side evaluation, and 5 stanzas are used for standalone evaluation.

We sample stanzas from the original data set to fill in the pairs for the side-by-side evaluation, plus 5 stanzas for the standalone evaluation. The stanzas from the original data set are tokenized in the same way as the input data for the poetry generation model, so they look similar. In total, each form has 15 pairs of stanzas for side-by-side evaluation, and 20 single stanzas for standalone evaluation. 120 generated stanzas, and 80 stanzas from the original NoRSC data set are evaluated by humans.

6.2.2 Results

Each form received between 6 and 7 answers. In total, 27 answer to the forms were submitted.

6.2.3 Rhyme ratings

The stanzas that got the best overall rhyme rating are the original NoRSC stanzas. They got an average rhyme rating of 2.63. The second best stanzas are those generated by the line based model, with an average rhyme rating of 2.27. For these stanzas, 3 was the most frequent rating (see Figure 6.2). The

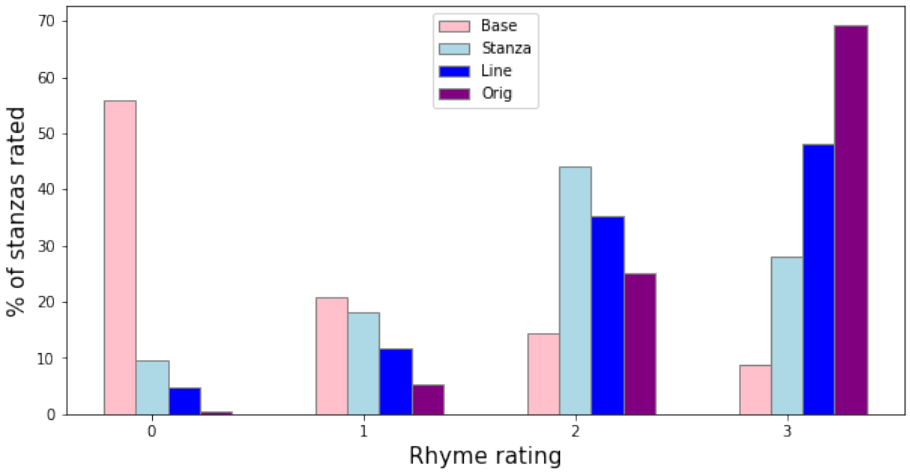

Figure 6.2: Frequency of rhyme rating for the stanzas

stanzas produced by the stanza based model got an average rhyme rating of 1.91, and were most frequently rated 2. The baseline model, which was simply the stanza based model without the rhyme-component, produced the least rhyming stanzas, with an average rating of 0.76, and 0 being the most frequent rhyme rating.

From these data we can conclude that both the line based and stanza based poetry generation models were able to model rhyme better than the baseline model, but not as well as the original data. The line based poetry generation performed significantly better than the stanza based poetry generation model.

6.2.4 'Written by a human'-ratings

Across all models, the generated poetry is more often rated 'written by a human' in the standalone evaluation than in the side-by-side. This was expected, and is also seen in Nikolov et al. (2020).

In the side-by-side evaluation, the original NoRSC stanzas were rated 'written by a human' 85.19% of the time (see Table 6.2). This means the evaluators combined got an accuracy of 85.19%, and that they in 14.81% of the cases annotated a generated stanza as written by human.

In the side-by-side evaluation, model that most often was rated 'written by a human' is the baseline model, as 19.26% of the ratings for the stanzas generated by the baseline model were 'written by a human'. The stanzas generated by the line based model were rated to be written by a human 17.78% of the time, and the stanzas from the stanza based model 7.41% of the time (see Table 6.2).

In the standalone evaluation, the original NoRSC stanzas were rated 'written by a human' in 82.96% of the ratings. This is a slight decrease from the side-by-side evaluation. This can indicate that it is easier to decide if a stanza is "real" if it is presented in a pair.

Stanzas	% of times rated to be written by a human in side-by-side evaluation	% of times rated to be written by a human in stand-alone evaluation	% of times rated to be written by a human in total
NoRSC stanzas	85.19	82.96	84.63
Baseline model	19.26	25.19	22.22
Line-based model	17.78	48.89	33.33
Stanza-based model	7.41	11.11	9.26

Table 6.2: Frequency of 'written by a human'-ratings for the different types of stanzas in side-by-side and standalone evaluation

From the standalone evaluation, the best model is the line based model. The stanzas generated by the line based model are rated 'written by a human' in 48.89% of the ratings. The next best, with regards to being rated as written by a human, is the baseline model. Its stanzas were rated to be written by a human 25.19% of the time. The stanzas generated by the stanza based model were rated to be written by a human in 11.11% of the ratings.

6.2.5 The relation between rhyme and perceived "written by a human"-ness

We wanted to see if there was a correlation between the stanzas getting good rhyme scores and them being rated as written by a human. We cannot conclude that there is a direct connection, because the stanza based model scores higher than the baseline model when it comes to rhyme, but worse when it comes to being rated as written by a human. The stanza based model generally performs worse than expected. It is surprising that the rhyme scores for the stanza based model are that much lower than the line based model, considering that they use the same rhyme model.

One possible reason for this is a the way the stanza based poetry generation model works. Recall section 6.1.2 where we describe the generation algorithm for the stanza based model. The line-ending words are sampled from the rhyme model. These are inserted into the sequence during generation, which are then sent back to into the language model. Because the line-ending words are selected solely based on the rhyme models probabilities, this can lead to word sequences that are very unlikely according to the language model.

But this cannot be the only reason. See the stanzas 6.1 and 6.2. In both, the last line looks very weird, with too many words bunched together with little to no grammar. But since the stanzas are generated backwards, we cannot blame this on unlikely words being inserted. When these lines were generated, no words (except for the very last, the prompt that started generation) had been

inserted.

(6.1) fjellene i storfangst går og fuglesang
hvis morgenglans er klokkeklang
lukk mitt nordlys jøde kastes begynte stamsund flod nærmer sang

(6.2) russlands sære kogleri
jeg kom mine mål til å eie
og det er en gyllen dag så bred
han føler tåker og av deres kunster
for norges liv
norske norske norske norske norske hunger grønnes varslet mumler urørt
urørt favntak betenk bitterhet slemt karpus grisjka løvspring bakved 1942

Another thing that is interesting about 6.1, is that though all lines in the stanza seems to rhyme, it has consistently been rated 2 more often than 3. It might be the case that if the structure of the poetry is bad, they are not perceived to rhyme as well. We need to make further experiments before we can conclude any of these suggestions.

6.2.6 Controversial stanzas

As mentioned, each form received 6-7 answers, meaning that every stanza has been evaluated by 6-7 different people. Some stanzas have very disagreeing ratings. For example, there are 25 stanzas where there are at least 3 people that annotated the stanza to be written to be human, *and* at least 3 people that annotated the stanza to *not* be written by a human. Below are some of these stanzas:

(6.3) med yndig majestet han går
som gjorde sinnet blått av romantikk
vansiret av laster sykdom og sår
plutselig stod stauper igjen for mitt blikk

(6.4) skurken saulus løfter armen
atter dette stikk i barmen
denne angst de syner røde
taus han stirrer på den døde

(6.5) nuets underveis
se det er det jovisst
der ligger så stille mot vest hvor sol går ned
som klippene og sluttet kvist

(6.6) så må de ha vise
kan trenge seg inn
nå er den å farvel farvel
å farvel farvel
du som fulgte i somre atten

6.3 was generated by the line based poetry generation model. 6.4 is from the original data set. 6.5 was generated by the stanza based poetry generation model, and 6.6 was generated by the baseline model.

As for controversial stanzas with regards to rhyme, there are 19 stanzas that have been given both a score of 0 and a score of 3 in rhyme rating. We list some of these below:

(6.7) hvor bårer med skum over utsteinen slår
og kjølens siste bakker i det kildested
det første hender med høstens redsler død
hin yngste apall strør blikke hevet goder tå flyver ned

(6.8) da lyner et lys ned i dypet
en gnistrende sprøyt av ild
det glitrer av blånende bølger
er havet likevel til

(6.9) inn i fjellets flammesky
en lovsangs brus som storm mot sky
fra sinnet på ny og på ny
fra sinnet på ny og på ny

(6.10) og tross den tross dødninghaven
over vennens liv er bange
hver gang han synger se ham
den er seirens dag til gry

6.7 was generated by the stanza based poetry generation model. It received the rhyme scores [3, 2, 2, 2, 2, 1, 0]. 6.8 is from the NoRSC data set. It received the rhyme scores [3, 3, 3, 2, 2, 2, 0]. 6.9 was generated by the line based poetry generation model, and received rhyme scores of [3, 3, 3, 3, 3, 3, 0]. 6.10 was generated by the baseline model, and received rhyme scores of [3, 1, 1, 0, 0, 0, 0].

It seems that for most of the controversial stanzas with regard to rhyme score, there is only one score of 3 or 0, and the rest of the rhyme scores are more or less in agreement. This is the case for examples 6.8, 6.9 and 6.10. In these cases, the cause for the score might have been a mis-click made by one evaluator. Example 6.7, on the other hand, received all four possible rhyme scores, so it is not as easy to just dismiss this as mis-clicks. Still, the most frequent score for this stanza, 2, is the appropriate annotation according to the instructions.

6.2.7 Original stanzas rated 'not written by a human'

There were three stanzas from the NoRSC data set that received more 'not written by a human' ratings than 'written by a human' ones. All these three were from the side-by-side evaluation, which means that it is not just the quality of the original stanza, but also the quality of the stanza it was paired with, that contributes to the ratings.

The stanza in 6.11 was annotated written by a human 2 times, and not written by a human 5 times. The stanza it was paired with, 6.12, was generated by the line based poetry generation model. Interestingly, though 6.11 is follows a solid ABAB rhyme pattern, it received the rhyme ratings [3, 3, 2, 2, 2, 1, 0]. 6.12 also follows the same rhyme pattern, and this was rated slightly higher with the rhyme scores [3, 3, 2, 2, 2, 1, 1].

(6.11) tillykke med dåden dere frelste imperiet
mac donald og thomas og henderson
og jobben var hård dere fortjener en ferie
men først må dere kysse hr. baldwins hånd

(6.12) han er jo
hva var det en guds fiolin
templet du har skjendet med vold og blod
især i det siste de trenger seg inn

The stanza in 6.13 was annotated written by a human 3 times, and not written by a human 4 times. It was paired with 6.14, which was generated by the line based poetry generation model. The original stanza received perfect rhyme scores from all evaluators.

(6.13) det følte hver som kom
fra reis igjen og så seg om
det følte hver som gikk
i siste avskjedsblikk

(6.14) sjelevingens hvilegren
har båret det ord og det ble igjen
frem stormer egyptens armé
et dryssende stjernegry
fra sinnet på ny og på ny
hans veier ble lys for lys og fred

The stanza in 6.15 was annotated written by a human 3 times, and not written by a human 4 times. It was paired with 6.16, which was generated by the baseline poetry generation model. This example also demonstrates that a stanza rhyming does not make seem more like it is written by a person, as 6.15 does not rhyme at all. It makes up for the lack of rhyme by being surprisingly coherent and grammatical.

(6.15) å våke over fedrelandets lover
å sørge for at ingenting forlises
er smukt betryggende for folkehellet
men det er skjønnere med ånd å våge
å slå med moses tryllestav på fjellet
skjønt nytten ei kan fattes og bevises

(6.16) men se han kommer med venner
på klippens bryst da lød
evangeliets herlige bud