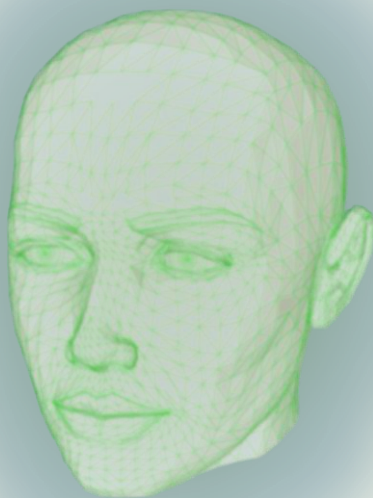


# The Friendly Poetry Hacker



## Hvad og Hvordan

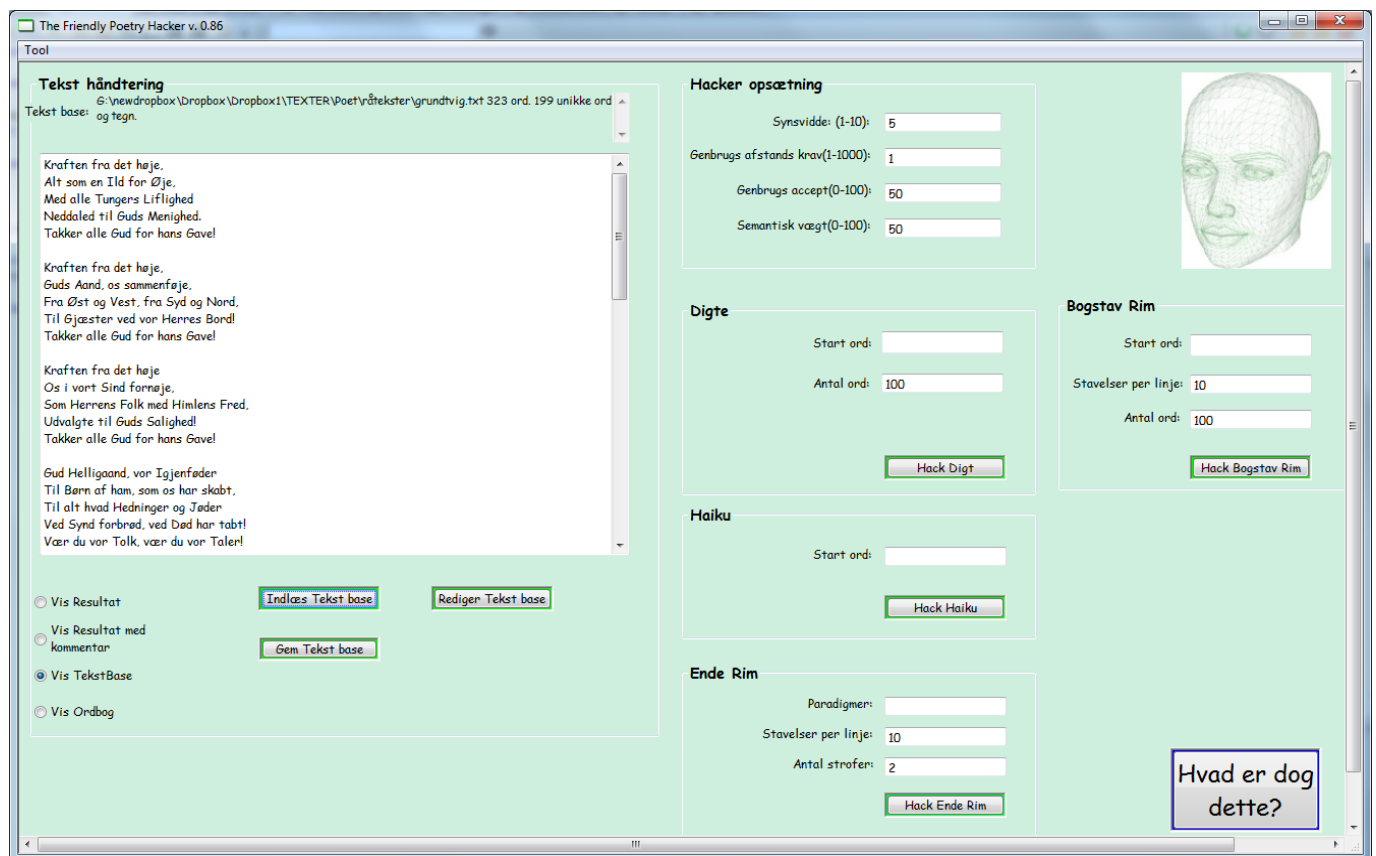
Smilende trafikofre kredser  
husvilde undertiden  
efterladte jeg kan leder  
få mit smil af undertiden.

Bare hænge med  
skuffen eller skrålende  
modtog jeg dåben.

## Indhold

Introduktion.....	3
Brugervejledning.....	4
At få fat i en basistekst:.....	4
Du vælger hacker indstillinger: .....	4
Justeringer .....	4
Gode råd .....	5
Manual .....	6
Tekst håndtering. ....	6
Hacker opsætning .....	9
Digte.....	10
Haiku.....	11
Enderim opsætning.....	11
Bogstavrim .....	11
Hvad er dog dette?.....	12
Virkemåde .....	12
Teknik .....	12
Haiku teknik .....	12
Bogstavrim teknik .....	12
Enderim teknik .....	12
Tegn.....	13
Hvorfor? .....	13
Fremtid .....	13
Fejl .....	13

## Introduktion



*The Friendly Poetry Hacker* analyserer en tekst, 'basisteksten' og danner en ny tekst, som ligner den analyserede i stil og indhold.

Hvad ved *The Friendly Poetry Hacker* efter den har analyseret basisteksten?

Den ved, hvilken rækkefølge ordene optræder og deres hyppighed.

Den ved, hvilke stavelser der optræder i ordene.

Den ved, hvor der følger linjebrud og kommatering efter ordene i basisteksten.

Hvad gør man for at få *The Friendly Poetry Hacker* til at syntetisere en tekst?

Man beder *The Friendly Poetry Hacker* om at indlæse en basistekst.

Man giver *The Friendly Poetry Hacker* et start ord.

Man beder *The Friendly Poetry Hacker* om at danne en ny tekst.

Man beder om at gemme den nye tekst i et dokument.

*The Friendly Poetry Hacker* er en app. Der kan køre på en Pc med Windows XP eller senere.

Den er testet på MacBook med macOS 10:7:5 Snowleopard og macOS Yosemite 10:10:5.

Copyright: Søren Skogstad Nielsen  
email: [skogstadnielsen@gmail.com](mailto:skogstadnielsen@gmail.com)

## Brugervejledning

Dette afsnit fortæller, hvordan man gør. *The Friendly Poetry Hacker* er et værktøj på samme måde, som en kuglepen er et værktøj. Så dette afsnit er en vejledning i, hvordan man bruger *The Friendly Poetry Hacker* til at lave en almindelig tekst, haiku, enderim, eller bogstavrim ud fra basisteksten.

### At få fat i en basistekst:

Hvor kan man finde basis tekster? Masser af steder på nettet. F.eks: <https://kalliope.org/> Kig også i Blogs mv. I folderen 'råtekster' findes nogle teksteksempler.

Du flytter den tekst, du vil bruge som basis tekst over i et dokument uden formatering altså i et .txt format. De fleste tekstbehandlere kan det via der deres gemdokument facilitet.

Eller:

Du trykker på 'Rediger Tekst base' knappen. Det åbner redigeringsvinduet, og du sætter din basistekst ind her ved cut/paste og trykker 'ok' knappen . Så er teksten klar til at eksperimentere med. Du kan også højre klikke i redigeringsvinduet, så får du en popup menu der kan indsætte teksten.

### Du vælger hacker indstillinger:

Anbefalede til at starte med:

- Synsvidde: 5
- Genbrugs accept: 50
- Genbrugs afstands krav: 1
- Semantisk vægt: 50

## Justeringer

Er der for mange genbrugte ord i den syntetiserede tekst?

Nedsæt 'genbrugs accept' f.eks. fra 50 til 10.

Og øg måske 'genbrugsafstanden' til f.eks. 10.

For at et ord kan blive genbrugt i tekstsyntesen, skal det nemlig passere to barrierer. Den første er, at der skal være mere end 'Genbrugsafstand' ord positioner tilbage til, hvor det sidst blev brugt. Den anden barriere er en lodtrækning, hvor barrierens lod gives en tilfældig talværdi mellem 0 og 100. Og ordets lods værdi er det, du angiver som genbrugsaccept. Jo højere genbrugsacceptværdien er, jo større chance har ordet for at vinde over barrieren og derved for at blive genbrugt.

Sættes 'Genbrugs accept' høj, får *The Friendly Poetry Hacker* nemmere ved at fiske ord frem i tekstsyntesen. Derved kan den danne større tekststykker uden at den løber tør for brugbare ord. Men det kan også hænde, at *The Friendly Poetry Hacker* går i 'løkke' og danner tekster, der består af f. eks.:

*i og en og i og en og i og en og i og en og i og en og i og en...*

Det kan man rette op på ved at sætte 'Genbrugs afstands krav' til en højere værdi.

Genbrug er i princippet en god ting, digtere belyser ofte et bestemt begreb, ved at gentage det i nye sammenhænge. Så ved at eksperimentere lidt med de to genbrugs regulatorer kan spændende nye ordsammenstillinger lokkes frem.

Øges 'Synsvidde' bliver *The Friendly Poetry Hacker* bedre i stand til at opfange længere strukturer i teksten, hvis de findes. Op til flere linjer i basisteksten kan blive gendannede. Sættes 'Synsvidde' til 1 vil POHAC ikke vide andet end, hvad næste ord efter et bestemt ord bør være.

Har *The Friendly Poetry Hacker* en tendens til at foretrække korte ord som 'den' 'jeg' 'sorg' 'han'. Så øg den semantiske vægt til f.eks. 70. Er der omvendt en tendens til, at *The Friendly Poetry Hacker* vælger korte ord fra, så nedsæt semantisk vægt til f.eks. 30.

Synes du, at du har brug for lidt mere indsigt i de beslutninger *The Friendly Poetry Hacker* har taget under tekstsyntesen, prøv da at vælge den runde knap: 'Vis resultat med kommentar'.

### Gode råd

Når du har lært *The Friendly Poetry Hacker* lidt at kende, så kig i manual afsnittet og få ideer til, hvordan du får *The Friendly Poetry Hacker* til at hjælpe med lige det, du gerne vil undersøge.

I almindelighed vil tekstsamlinger med betydningsladede ord f.eks. HAIKU samlinger være gode at starte med. Artikler eller blogs hvor eksperter og politikere kloger sig kan give både inspirerende, provokerende, og skægge resultater...

Har du en digtsamling, som du har svært ved at få hul på, så prøv at lade *The Friendly Poetry Hacker* generere nogle tekster med digtsamlingen som basistekst, modifier indstillingerne, sammenlign *The Friendly Poetry Hackers* tekster med de originale og bliv inspireret...

Prøv at lade basisteksten være en blanding af to forskellige digtsamlinger for eksempel af Grundtvig og Halfdan Rasmussen

Bliver du frustreret når *The Friendly Poetry Hacker* i en ellers flot sætning skriver 'og' i stedet for 'denne'?

Så husk at *The Friendly Poetry Hacker* ved ganske, ganske lidt om grammatik og semantik.

Jeg foreslår at du så leger 'redaktør' og lapper lidt på *The Friendly Poetry Hackers* tåbeligheder

- *The Friendly Poetry Hacker* bliver ikke sur og protesterer ;-).

## Manual

I dette manual afsnit finder du en detaljeret forklaring på, hvordan du håndterer værktøjet, det er altså ikke en vejledning i, hvad man kan eller bør gøre, men en vejledning i hvilken effekt *The Friendly Poetry Hackers* indstillinger og knapper har.

### Tekst håndtering.

I denne ramme findes følgende:

#### *Tekstbase*

I det store tekstfelt vises den indlæste tekstbase. Her vil synteseresultatet og ordbogen også blive vist. Stammer tekstbasen fra et indlæst dokument, vises stien til dokumentet, som tekstbasen stammer fra.

#### *Vis Tekstbase:*

Når denne runde knap klikkes, vises tekstbasen.

#### *Vis Resultat:*

Når denne runde knap klikkes, vises synteseresultatet, hvis det er dannet.

#### *Vis Resultat med kommentar:*

Når denne runde knap klikkes, vises synteseresultatet med kommentarer, hvis det altså er dannet.

Eks: på resultat med kommentarer:

*havde hjemme i*

[4]klappeland, *hjemme klappeland. hjemme klappeland.*

[13]klappeland, *alle de som klappe - ukomplet -*

#### *Kommentar:*

*Jeg genbrugte ikke: "~Cr" i position: 7*

*Jeg genbrugte:"hjemme" i position: 7*

*Jeg genbrugte:"klappeland" i position: 8*

*Jeg genbrugte ikke: "~Cr" i position: 10*

*Jeg genbrugte ikke: "klappeland" i position: 10*

*Jeg genbrugte:"hjemme" i position: 10*

*Jeg genbrugte ikke: "~Cr" i position: 11*

*Jeg genbrugte:"klappeland" i position: 11*

*Jeg genbrugte ikke: "klappeland" i position: 12*

Copyright: Søren Skogstad Nielsen  
email: [skogstadnielsen@gmail.com](mailto:skogstadnielsen@gmail.com)

"klappeland " udeladt, ville optræde dobbelt i position: 12

[4] og [13] ovenfor er ordpositionerne i den dannede tekst.

### Vis Ordbog:

Når denne runde knap klikkes, vises den ordbog som er dannet under analysen i følgende sammentrængte layout:

```
<Ordbog>::=<record>*
<record>::=<nøgleværdi> : <positionsfelt>*
<positionsfelt>::= | <positionsnummer> <positionsliste>*
<positionsliste>::=<ord i position>[<antal gange ordet optræder i positionen>]
```

Eksempel:

har : | 1 *hjemme*[2] | 2 *i*[2] | 3 *klappeland*[1] ~Cr[1] | 4 *klappeland*[1] .[1] | 5 ~Cr[1] ,[1]

Forklaring på eksemplet:

har :	<nøgleværdi> :	Forklaring
1	<positionsnummer>	Alt frem til næste   vedrører position et efter <i>har</i> .
<i>hjemme</i>	<ord i positionen>	<i>hjemme</i> optræder i første position efter <i>har</i> i basisteksten
[2]	<antal gange ordet optræder i positionen>	<i>hjemme</i> optræder to gange på første position efter <i>har</i> i basisteksten
2	<positionsnummer>	Alt frem til næste   vedrører anden position efter <i>har</i> .
<i>i</i>	<ord i positionen>	<i>i</i> optræder i anden position efter <i>har</i> i basisteksten
[2]	<antal gange ordet optræder i positionen>	<i>i</i> optræder to gange i anden position efter <i>har</i> i basisteksten
3	<positionsnummer>	Alt frem til næste   vedrører tredje position efter <i>har</i> .
<i>klappeland</i>	<ord i positionen>	<i>klappeland</i> optræder i tredje position efter <i>har</i> i basisteksten
[1]	<antal gange ordet optræder i positionen>	<i>klappeland</i> optræder en gang i tredje position efter <i>har</i> i basisteksten
~Cr	<ord i positionen>	~Cr ( <i>linieskift</i> ) optræder i tredje position efter <i>har</i> i basisteksten
[1]	<antal gange ordet optræder i positionen>	~Cr optræder en gang i tredje position efter <i>har</i> i basisteksten.
4	<positionsnummer>	Alt frem til næste   vedrører fjerde position efter <i>har</i> .
<i>klappeland</i>	<ord i positionen>	<i>klappeland</i> optræder i fjerde position efter <i>har</i> i basisteksten
[1]	<antal gange ordet optræder i positionen>	<i>klappeland</i> optræder en gang i fjerde position efter

Copyright: Søren Skogstad Nielsen  
email: [skogstadnielsen@gmail.com](mailto:skogstadnielsen@gmail.com)

	optræder i positionen>	<i>har</i> i basisteksten.
.	<ord i positionen>	. ( <i>punktum</i> ) optræder i fjerde position efter <i>har</i>
[1]	<antal gange ordet optræder i positionen>	. optræder en gang i fjerde position efter <i>har</i> i basisteksten.
5	<positionsnummer>	Alt frem til næste   vedrører femte position efter <i>har</i> .
~Cr	<ord i positionen>	~Cr ( <i>linieskift</i> ) optræder i femte position efter <i>har</i> i basisteksten
[1]	<antal gange ordet optræder i positionen>	~Cr ( <i>linieskift</i> ) optræder en gang i femte position efter <i>har</i> i basisteksten.
,	<ord i positionen>	, ( <i>komma</i> ) optræder i femte position efter <i>har</i> i basisteksten
[1]	<antal gange ordet optræder i positionen>	, optræder en gang i femte position efter <i>har</i> i basisteksten.

#### *Indlæs Tekst base:*

Knappen åbner et vindue, der viser dokumenterne på harddisken. Du udvælger det dokument, hvor den basistekst ligger, som du ønsker at bruge. Teksten vil blive indlæst og overskrive den basistekst *The Friendly Poetry Hacker* allerede har. Du kan også bruge 'cut' og 'paste' se [Rediger Tekstbase](#).

*OBS: The Friendly Poetry Hacker kan kun behandle dokumenter med ren tekst (.txt) format. :OBS*

#### *Rediger Tekst base:*

Knappen åbner et redigeringsvindue.

På en windows PC kan man redigere i vinduet via tastaturet med:

CTRL-C knap (kopiering af tekst).

CTRL-V knap (indsætning af tekst).

CTRL-X knap (fjerne markeret tekst)

Delete knap (fjerner markeret tekst eller tekst til højre for tekstkursoren).

Backspace knap (fjerner markeret tekst eller tekst til venstre for tekstkursoren).

Mac, har tilsvarende muligheder.

Højreklik menuen har de samme funktioner.

Når du lukker redigeringsvinduet med 'Ok' knappen, erstattes tekstbasen med teksten i redigeringsvinduet.

#### *Gem Tekst base:*

Knappen åbner en dialogboks, hvor du vælger det dokument, som tekstbasen skal gemmes i.



*Gem Resultat:*

Knappen åbner en dialogboks, hvor du angiver navnet på det dokument, det viste synteseresultat skal gemmes i.

**Hacker opsætning**

I denne ramme kan du ændre de parametre, der styrer *The Friendly Poetry Hackers* virkemåde.

*Synsvidde: (1-10):*

Feltet angiver, hvor mange ordpositioner *The Friendly Poetry Hacker* ser frem i basisteksten fra en bestemt ordposition.

Angiver du 1, ser *The Friendly Poetry Hacker* kun en position frem. Angiver du 10, ser den 10 positioner frem. Dette gælder for både tekstanalysen og tekstsyntesen.

*Genbrugs accept(0-100):*

Feltet angiver sandsynligheden for, at *The Friendly Poetry Hacker* i tekstsyntesen vil bruge et ord fra tekstbasen, som den allerede har brugt.

Angiver du 0 vil *The Friendly Poetry Hacker* aldrig genbruge et ord.

Angiver du 100 vil *The Friendly Poetry Hacker* ikke betænke sig på at genbruge ethvert ord, der overholder *Genbrugs afstands krav*.

*Genbrugs afstands krav(1-1000):*

Feltet angiver hvor mange mellemliggende positioner, der skal være fra sidste brug af et bestemt ord, før *The Friendly Poetry Hacker* i tekstsyntesen må genbruge det.

Angiver du 0 vil *The Friendly Poetry Hacker* genbruge ordet, hvis det også kan passere filteret angivet i *Genbrugs accept*.

*Semantisk vægt(0-100):*

Denne flotte vending dækker over den vægt, et ord får, når der i syntesen er flere ord, der optræder med samme hyppighed i en bestemt position. Semantisk vægt styrer, hvor stor en vægt antal bogstaver i et ord får ved udvalget af et ord blandt de ord, som analysen har vist, kan optræde i en bestemt position.

Skriver du 0, får ordlængden af et ord, blandt de ord der kan optræde i en bestemt position ingen betydning, det er alene hyppigheden af ordet, som bestemmer.

Skriver du 100, får hyppigheden af forekomsten af et ord i en bestemt position ingen betydning, ved valget mellem mulige ord.

Skriver du 50 så vil ordlængde og ordhyppighed indgå med samme vægt.

Vægtningssformlen ser sådan ud:

$\text{vægt} = \text{ordlængde} * \text{semantisk vægt} + (100 - \text{semantisk vægt}) * \text{hyppighed}$

Eksempel:

I en bestemt position viser analysen at følgende ord er mulige:

Ord	Hyppighed	Ordlængde
i	10	1

protokol	2	8
tempo	7	5

Med en semantisk vægt på 100 vil 'protokol' blive udvalgt på grund af ordlængden.

Ord	Vægt
i	100
protokol	800
tempo	500

Med en semantisk vægt på 0 vil 'i' blive udvalgt på grund af hyppigheden.

Ord	Vægt
i	1000
protokol	200
tempo	700

Med en semantisk vægt på 50 vil 'tempo' blive udvalgt, selvom det hverken er det hyppigste eller længste ord blandt de mulige.

Ord	Vægt
i	550
protokol	500
tempo	600

## Digte

Digte angiver en tekstsyntese, hvor den dannede tekst ikke er underlagt andre formkrav, end de der er nævnt i **Hacker opsætning**. *The Friendly Poetry Hacker* har brug for, at du angiver et startord, og hvor mange ord tekstsynteseresultatet skal indeholde.

### Start ord:

Det ord som teksten skal starte med. Ordet skal findes i ordbogen, og derfor også i basisteksten. Specielt kan ordet godt være et tegn f.eks. et punktum eller ~Cr (Linjeskift)

### Antal ord:

Her angives det antal ord, som den tekst, der syntetiseres, skal indeholde. Feks. 100.

### Hack Digt

Når denne knap trykkes, startes tekstsyntesen. Den vil blive vist i tekst feltet i teksthåndterings rammen.

## Haiku

*The Friendly Poetry Hacker* kender til følgende HAIKU konvention:

Teksten skal bestå af tre linjer, Første linje skal indeholde 5 stavelser, næste linje skal indeholde 7 stavelser. Sidste linje skal indeholde 5 stavelser.<sup>1</sup>

Der er en del andre haiku regler, mest semantiske, dem kender *The Friendly Poetry Hacker* ikke.

### Start ord:

Det ord som haikuet skal starte med. Ordet skal findes i ordbogen, og derfor også i basisteksten.

### Hack Haiku

Når denne knap trykkes, startes dannelsen af et haiku. Det vil blive vist i tekst feltet i teksthåndteringsrammen.

Note: I HAIKU syntesen vil alle tegn og linjeskift blive ignorerede.

## Enderim opsætning

### Paradigmer:

Her skrives rim paradigmer i form af et eller flere ord, hvis slutstavelser der skal findes rim til i tekstsyntesen. Ordene behøver ikke at findes i basisteksten. Adskil hvert af ordene med et mellemrum. For hvert paradigme vil der blive dannet to linjer.

### Stavelser per linje:

Hver linje i de dannede strofer vil få det antal stavelser, som er angivet her. Hvis det er muligt med de ord, der er til rådighed i positionerne.

### Antal strofer:

Her angives, hvor mange strofer den dannede tekst skal bestå af.

### Hack Ende Rim

Når denne knap trykkes, startes dannelsen af en enderimstekst. Resultatet vil blive vist i tekst feltet i teksthåndteringsrammen.

## Bogstavrim

### Start ord:

Dette ords startbogstav angiver, hvilket startbogstav alle ord i bogstavrimet skal starte med. Startordet skal findes i ordbogen. Tegn er i princippet tilladte som startord, men da ingen ord starter med et tegn, vil *The Friendly Poetry Hacker* ikke kunne danne en syntesetekst

---

<sup>1</sup> <https://da.wikipedia.org/wiki/Haiku>

### Stavelser per linje:

Hver linje i den dannede tekst vil få det antal stavelser, som er angivet her.

### Antal ord:

Antal ord der skal være i den dannede tekst.

### Hack Bogstav Rim

Når denne knap trykkes, startes dannelsen af en bogstavrimtekst. Resultatet vil blive vist i tekst feltet i teksthåndteringsrammen til venstre.

### Hvad er dog dette?

Vil vise den tekst du læser nu.

## Virkemåde

### Teknik

I sin analyse af basisteksten anvender *The Friendly Poetry Hacker* superpositionerede markovkæder suppleret med regler for opdeling af ord i stavelser. Denne viden anvender *The Friendly Poetry Hacker* dernæst i tekstsyntesen, når den skal beslutte hvilke ordsekvenser der bør følge efter et bestemt ord.

### Haiku teknik

Ved under analysen ignoreres tegn og tal.

Opbygningen af en haiku linje sker på samme måde, som ved syntesen af en almindelig tekst, linje for linje. Men ved valget af næste ord, der skal tilføjes til linjen, overholdes formkravet om antal stavelser per linje (5 - 7 - 5). Parametrene i [Hacker opsætningen](#) bliver anvendt ved valget af ord.

### Bogstavrim teknik

Opbygningen af en linje i et bogstavrim sker på samme måde, som for en almindelig tekst linje for linje. Dog, hvis tekstsyntesetabellen ikke indeholder ord med det rette startbogstav, så vælger *The Friendly Poetry Hacker* et ord fra ordbogen med et korrekt startbogstav. Linjeskift anbringes, når det angivne antal stavelser i linjen er nået, eller måske er overskredet af sidste ord. Der gøres ikke forskel på store og små bogstaver ved syntesen af bogstavrim.

### Enderim teknik

Enderimsyntesen starter bagfra for hver linje. Her indsættes et ord, hvis sidste stavelse rimer med sidste stavelse i første paradigme. Næste paradigme bestemmer sidste ord i næste linje. Og processen gentages en gang til. To paradigmer giver altså fire linjer. Når alle paradigmer er brugt to gange afsluttes strofen med en blank linje og en ny strofe startes. Valget af næstsidste ord og de foranstående sker ud fra markov kæderne, men ved enderim arbejdes baglæns når linjen syntetiseres.

### Rimteknikker

Når *The Friendly Poetry Hacker* skal afgøre om en stavelse rimer med en anden stavelse så bruger den følgende regler:

- De to stavelser skal hver indeholde en vokal, og de to vokaler skal være ens.
- Følger der et bogstav efter vokalen, skal det være det samme bogstav i begge stavelser.

Når *The Friendly Poetry Hacker* afgør om et ord rimer mere eller et mindre med et andet sammenligner den stavelserne par for par bagfra. Jo flere rimene stavelser den finder, jo bedre synes POHAC at de to ord rimer. Sammenligningen har 'bagvægt', så de sidste stavelser tæller mest.

#### Note:

*Bedre forslag til bestemmelse af om stavelser 'rimer' modtages med glæde og fornøjelse. Send en mail: [skogstadnielsen@gmail.com](mailto:skogstadnielsen@gmail.com)*

### Tegn

POHAC mener at følgende alle er tegn:

' ; , . ( ) [ ] ? ! " (Linieskift)

Alt andet mener *The Friendly Poetry Hacker* er bogstaver der er en del af et ord.

### Hvorfor?

Grunden til *The Friendly Poetry Hacker* er lavet, og har så mange muligheder er at jeg selv skriver og læser digte, og at jeg derigennem er blevet opmærksom på, hvordan sætningers ordsammenstillinger smitter af på ordenes semantik, og at det sommetider kan give helt nye oplevelser, når ord der normalt ikke finder sammen, pludselig gør det. *The Friendly Poetry Hacker* ved jo ikke, hvilke ord der bevidst plejer at blive sat sammen af en forfatter. Men *The Friendly Poetry Hacker* kan i sin analyse se, hvad forfatteren gør i praksis og på den måde skyde sig ind på bestemte temaer – komme på sporet af 'en stil'. Det er interessant.

### Fremtid

Det er min tanke at *The Friendly Poetry Hacker* skal blive dygtigere i flere retninger. I hvert fald bedre til stavelser, til rimdannelse, og den er ved at lære jamber, og daktyler, etc. at kende.

En anden ambition er at lære *The Friendly Poetry Hacker* at skrive sonetter.

Du må meget gerne bidrage. En version der kan fremmedsprog måske?

### Fejl

Opdager du at *The Friendly Poetry Hacker* laver en fejl, så send en mail om fejlen til [skogstadnielsen@gmail.com](mailto:skogstadnielsen@gmail.com) Ellers bliver fejlen næppe rettet.