

# MORETRAN

DAN BRUMLEVE

## 1. OVERVIEW

Moretran is a language for training a neural network with text.

A moretran script is a list of rules separated by newlines. The simplest kind of rule is two lists of words separated by “->”, for example:

```
hello how are you -> hi i am fine
goodbye -> bye
```

A special word is "\*" which is substituted with a random word during training, giving it the effect of a wildcard:

```
* is for dinner -> yum
```

The first clause of the rule can be a semicolon-separated list, representing the history of a thread:

```
whats your name ; * whats yours -> bob
* whats yours -> whats my what
```

The second clause of the rule can also be a semicolon-separated list, representing multiple outputs:

```
tell me a joke -> okay ; what does the fox say
```

A rule can be prefixed by a list of tags and a “:”. Tags can be used to represent different actors or properties, for example:

```
bob : whats your name ; * whats yours -> bob
sam : whats your name ; * whats yours -> sam
bob : whos your best friend -> sam
sam : whos your best friend -> bob
```

A rule can be weighted by appending “x” and a number to the arrow. This causes it to be entered into the training bag that many times. Example:

```
* ->x4 i dont understand
```

Comments are marked by “#”:

```
* ->x4 i dont understand # fallback response
```

A trailing “\” denotes a line continuation:

```
* -> all i have to say about the matter is blah \
    blah blah and i completely agree
```

That’s all there is to the language. A moretran script is a list of rules. Each rule is compiled to shibboleths. The positions of the wildcards within the clauses are also included, allowing substitutions to be performed. The neural network is then trained from the compiled moretran code.

## 2. TOOLS

moretran scripts can be compiled with the moretran utility. Example:

```
$ moretran -o test.more test.txt
```

## 3. EXAMPLE

**Example 1.** This moretran script defines the behavior of a bully. If you send him a message, he responds with a threat, and initiates a countdown. The countdown is interrupted if you pay him, otherwise he vandalizes your profile image. The substitution variables “\$nom”, “\1”, “\2”, etc. are interpreted by the application, not moretran itself.

```
# threaten
bully : from * * -> \
    tag $nom bullying ; \
    to \2 gimme money or get a black eye ; \
    count \2 0
# count
bully bullying : count * 0 -> \
    to \2 imma count to three ; \
    count \2 1
bully : count * 0 ->
bully bullying : count * 1 -> \
    to \2 one pay me ; \
    count \2 2
bully : count * 1 ->
bully bullying : count * 2 -> \
    to \2 two else imma hit you ; \
    count \2 3
bully : count * 2 ->
# vandalize
bully bullying : count * 3 -> \
    to \2 three thats it buddy ; \
    punch \2 ; untag $nom bullying
bully : count * 3 ->
# paid
bully bullying : paid by * * -> \
    untag $nom bullying ; \
    to \3 thanks buddy
```