Contents

1 Module Bred : main project

```
type output =
  | Stdout
  | Outfile of string
val string_of_output : output -> string
val learn : string -> string
val main : int -> 'a -> string list -> 'b -> unit
val deep : int Cmdliner.Term.t
val out : output Cmdliner.Term.t
val files : string list Cmdliner.Term.t
val num : int Cmdliner.Term.t
val main_t : unit Cmdliner.Term.t
val info : Cmdliner.Term.info
```

2 Module Read

```
val string_of_file : string -> string
```

3 Module Write

```
val generate_text : Markov.ptable -> string
```

4 Module Markov

```
type distribution = {
  total : int ;
  amounts : (string * int) list ;
```

```
}
type ptable = {
  prefix_length : int ;
  table : (string list, distribution) Hashtbl.t ;
}
val is_word : char -> bool
val is_punctuation : char -> bool
val is_sentence_separator : char -> bool
val split_word : string -> string list
val start : int -> string list
val shift : 'a list -> 'a -> 'a list
val add_to : ('a, 'b list) Hashtbl.t -> 'a -> 'b -> unit
val compute_distribution : string list -> distribution
val next_in_htable : ('a, distribution) Hashtbl.t -> 'a -> string
val build_ptable : string list -> int -> ptable
val walk_ptable : ptable -> string list
```