Цепи Маркова

И генерация текста

Базовые понятия

- **Случайная величина** X – величина, которая определяется результатом случайного явления (например, результат бросания кубика)
- **Случайный процесс** – набор случайных величин, проиндексированных множеством T, которое часто обозначает разные моменты времени.

Марковское свойство

Обычно, чтобы вычислить вероятность какого-то события в цепи событий, мы должны использовать **полную формулу вероятности**: перемножить все вероятности всех предыдущих событий.

Свойство отсутствия памяти: в любой момент времени условное распределение будущих состояний процесса с заданными текущим и прошлыми состояниями зависит только от текущего состояния, но не от прошлых состояний.

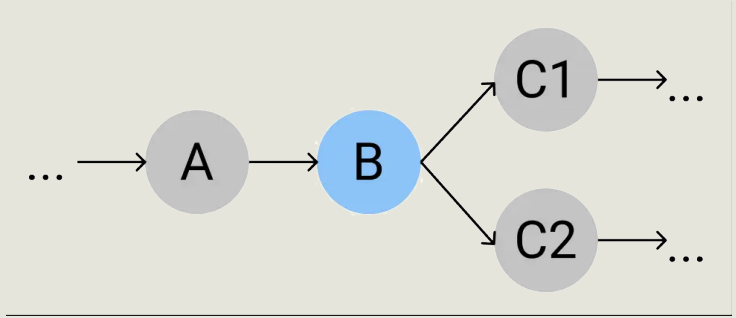
$$P(\text{future} \mid \text{present}, \text{past}) = P(\text{future} \mid \text{present})$$

Markov property

Цепь Маркова

Цепь Маркова — это последовательность событий, в которой наступление каждого события зависит только от предыдущего и не зависит от остальных событий.

В случае с генерацией текста событие – это появление следующего токена.

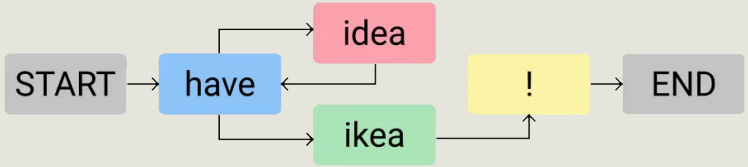


Генерация текста

Допустим, у нас есть такая последовательность токенов:

START → HAVE → IDEA → HAVE → IKEA → ! → END

В этой последовательности событий есть события, которые встречаются чаще других. Лучше это видно на графе переходов:



На таком графе показываются все возможные переходы между событиями.

Матрица переходов

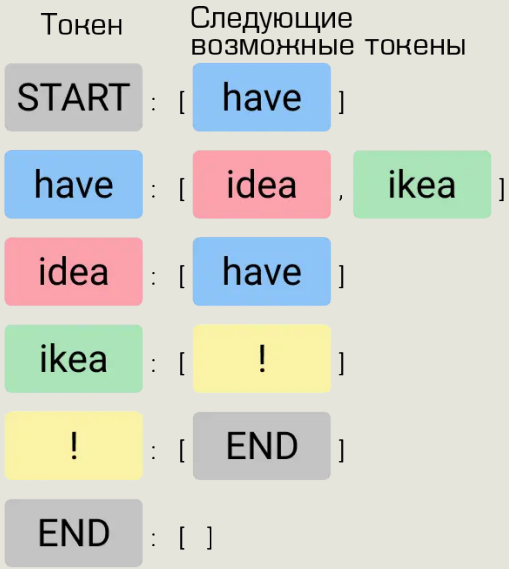
Матрица переходов – такая матрица, где в строках перечислены исходные состояния, а в столбцах — последующие. Если в ячейках отмечать вероятность перехода из исходного в последующее, то граф перехода нашей цепи мы можем представить в виде таблицы:

	START	have	idea	ikea	!	END
START	0	1	0	0	0	0
have	0	0	0.5	0.5	0	0
idea	0	1	0	0	0	0
ikea	0	0	0	0	1	0
!	0	0	0	0	0	1

Матрица переходов

Наша матрица – разреженная, в основном в ней содержатся нули – невозможные переходы. Способ записи можно сократить:

Событие	**Возможные последующие события**
START	→ have
have	→ idea, → ikea
idea	→ have
ikea	→ !
!	→ END



N-грамы

- Для генерации синтаксически правильного текста униграмов будет маловато: можно в качестве одного события использовать N–грамы (2, 3...)
- Но чем больше N, тем разреженнее вероятности