Трансформаторы. Токены

Безусловно, наиболее развитые методы трансформации текстов связаны с разработкой искусственных языков (большей частью, языков программирования), так что именно из этой области удобно заимствовать понятия и определения.

Два важных класса объектов, встречающихся в почти любом языке программирования, это константы и переменные.

Константа - это то, что представляет само себя. Ее значение всегда постоянно и не зависит от контекста. Фактически, это имя объекта, который не изменяется, либо изменения которого в данном контексте несущественны ("Big Ben", например).

В естественных языках, аналогами констант могут служить идиомы и устойчивые выражения: число и порядок слов в них фиксированы, а смысл часто невыводим из составляющих, как, например, во фразах: "сломя голову" или "скатертью дорога".

Значение переменной, наоборот, если и не полностью произвольно, то может меняться и зависеть от того, когда и где она встречена. Вплоть до того, что, во фразе: "Ну, да. Конечно" три утверждения подряд означают, на самом деле, отрицание.

При анализе алфавитного текста, одиночный символ является естественной, но часто бесполезной, мерой грануляции. Как видно из примеров выше, семантика основывается на более сложных структурах.

Разумеется, слово "символ" следует толковать расширительно - это может быть устойчивое графическое изображение (пиктограмма, например), устойчивая звуковая комбинация (музыкальная заставка), традиционные жесты ("взять под козырек", "сделать ручкой"), заклятия ("будь здоров", "не поминай лихом") и, в сущности, что угодно, чему назначен собственный смысл ("красная и зеленая ракеты - сигнал к атаке").

Поэтому, первой фазой преобразования текста, обычно, является токенизация: входной поток разбивается на группы алфавитных символов, которые сами являются символами другого языка (метаязыка). При этом, некоторые группы могут состоять из единственного алфавитного символа (возможно, имеющего иной смысл в метаязыке), а другие могут содержать произвольно длинные последовательности символов исходного алфавита.

Фактически, такое разбиение является Преобразованием Алфавита: текст, записанный символами исходного алфавита преобразуется в мета-текст, записанный символами мета-алфавита. На практике, цепочки подобных преобразований могут быть достаточно длинными. В сложившихся жаргонах, различные фазы обработки текста могут иметь различные собственные названия (такие как пре- и постпроцессинг, компиляция и линкование, лексический и синтаксический анализ, сжатие и шифрование итд).

Следует отчетливо понимать, что во всех случаях речь идет об одном и том же. В случае обратимых техник (сжатие, шифрование) не происходит потерь информации

при обработке, остальные процессы диссипативны - часть информации необратимо разрушается (аннотирование, реферирование, компиляция итп).

Методы автоматической структуризации текста крайне несовершенны и многие техники сжатия данных все еще базируются либо на обработке одиночных алфавитных символов (или иных блоков фиксированного размера), либо на простейших контекстных моделях (учитывающих внутренние зависимости, скажем, в пределах дюжины рядом расположенных алфавитных символов). Иногда эксплуатируются существенные особенности конкретного текста, например, разбиение на слова в естественных языках, тэги разметки в XML итд.

Статические методы сжатия явно разделяют фазы анализа и кодирования. В фазе анализа могут подсчитываться частоты токенов для энтропийного кодирования, например, или выполняться структуризация текста для создания словаря токенов.

Адаптивные методы, являются, по сути, "генетическими", в том смысле, что в процессе преобразования текста "выживают" только устойчивые комбинации.

Очевидно, что статические методы (подсчет "средней температуры по больнице") могут быть неэффективны или бесполезны для текстов переменного содержания, тогда как адаптивные, при удачном наборе параметров, могут быстро подстраиваться под динамический контент.