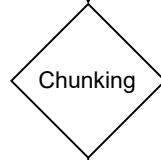


For our sketches we group multiple features together to form "super-features"

features very similar

Chunks that have one or more
(maximal values) in common are likely to be
similar, but small changes to the data are unlikely
to perturb the maximal values

Data



Generate R_FPS over
rolling window & make
permutation

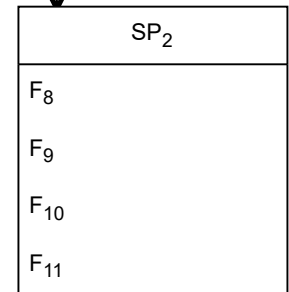
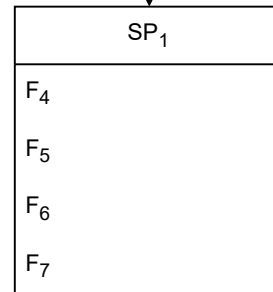
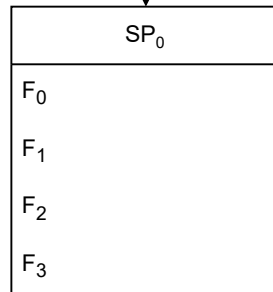
Chunks that have one or more
features (maximal values) in common are likely to be
very similar, but small changes to the data are unlikely
to perturb the maximal values

F₀: 0xff9ab74e
F₁: 0xff82845
F₂: 0xff1da26
F₃: 0xff7f156e
F₄: 0xff12c814
F₅: 0xffe3735c
F₆: 0xff32bd8e
F₇: 0xff9b8d2
F₈: 0xffe87e52
F₉: 0xff93729
F₁₀: 0xffe2fcdf
F₁₁: 0xff16ccf3

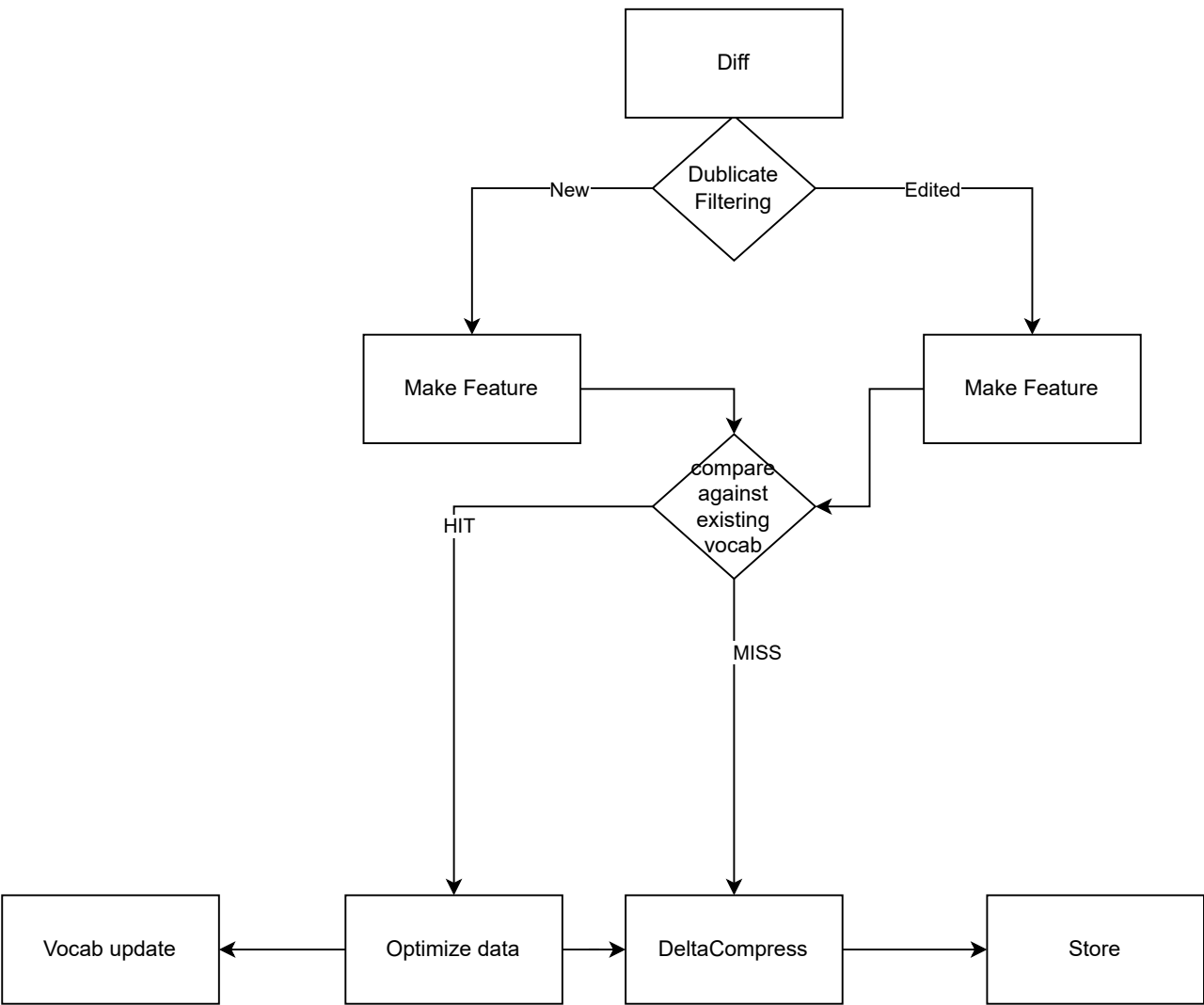
Making R_FPS for
interval of hashes to
obtain SP of that
interval

Increasing the number of
super-features increases
the number of matches

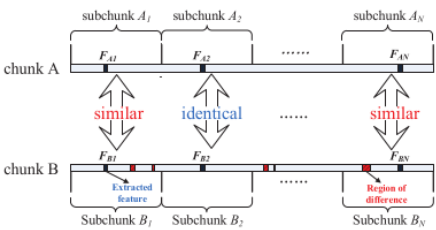
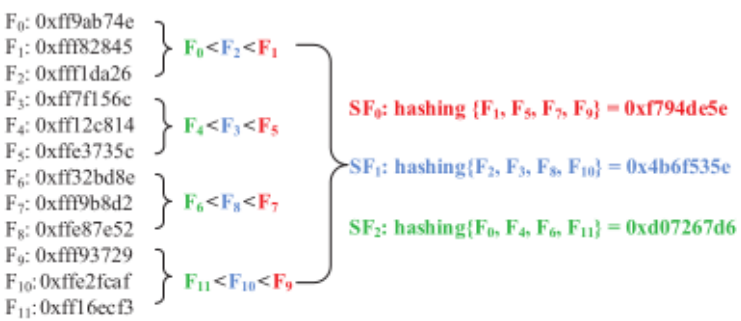
The super-feature value is a strong hash of the
underlying feature values. If two chunks have an
identical super-feature then all the underlying features
match.

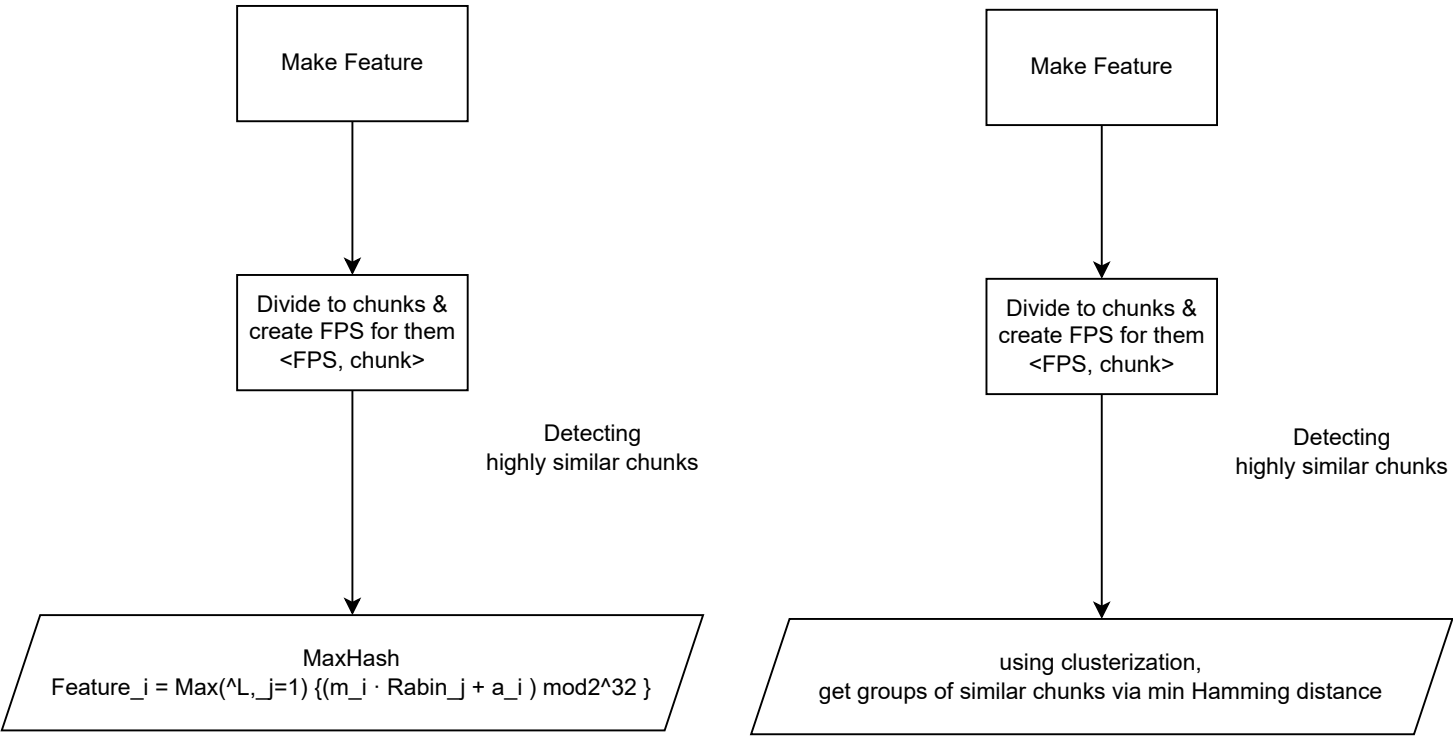


SuperFeature



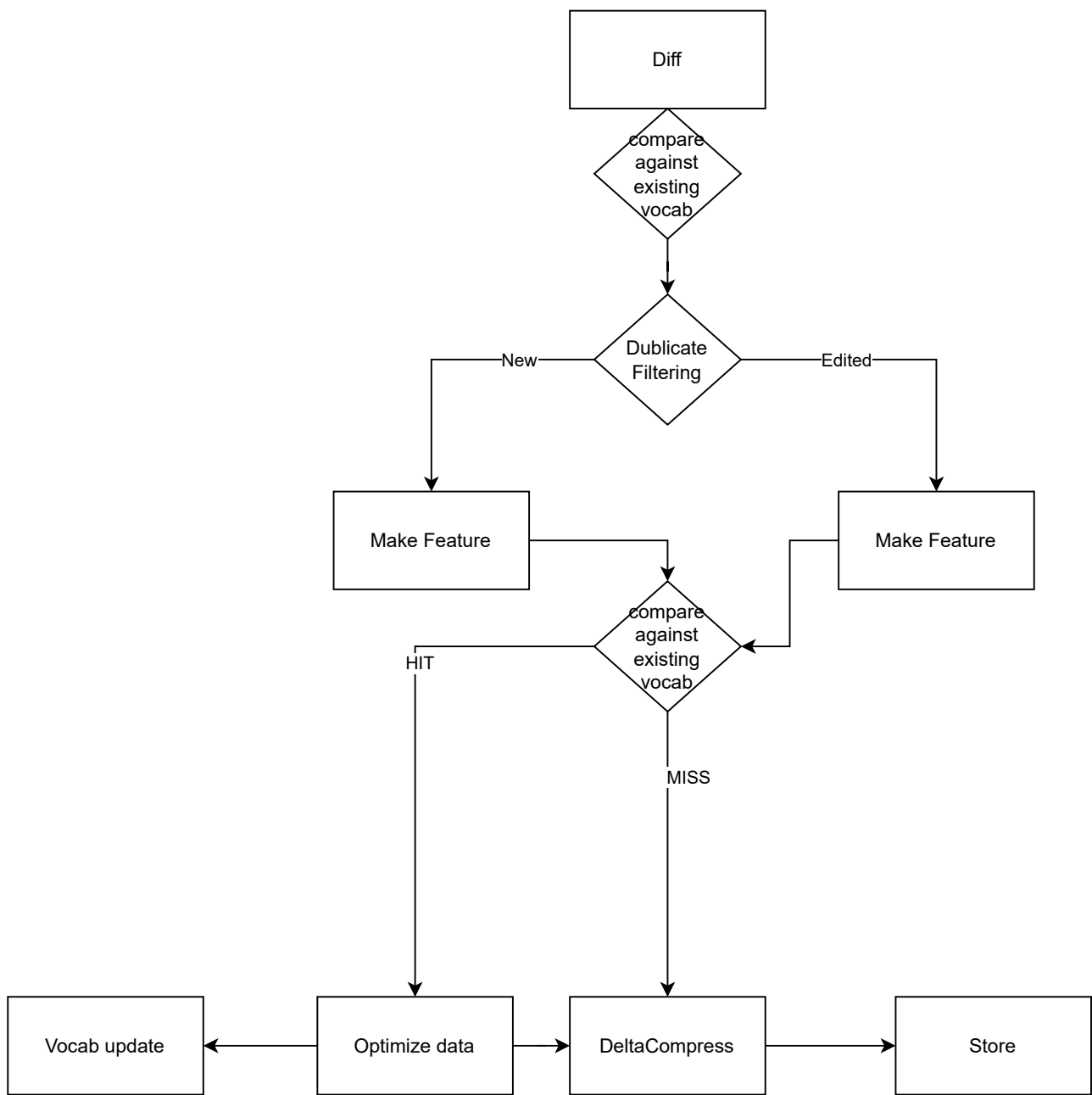
SS

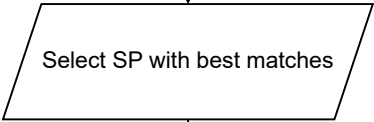




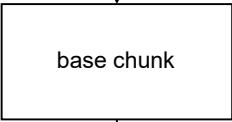
F₀: 0xff9ab74e
F₁: 0xffff82845
F₂: 0xffff1da26
F₃: 0xff7f156c
F₄: 0xff12c814
F₅: 0xffe3735c
F₆: 0xff32bd8e
F₇: 0xffff9b8d2
F₈: 0xffe87e52
F₉: 0xffff93729
F₁₀: 0xffe2fcaf
F₁₁: 0xff16ecf3

F₀: 0xff9ab74e
F₁: 0xffff82845
F₂: 0xffff1da26
F₃: 0xff7f156c
F₄: 0xff12c814
F₅: 0xffe3735c
F₆: 0xff32bd8e
F₇: 0xffff9b8d2
F₈: 0xffe87e52
F₉: 0xffff93729
F₁₀: 0xffe2fcaf
F₁₁: 0xff16ecf3

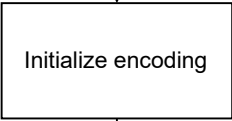




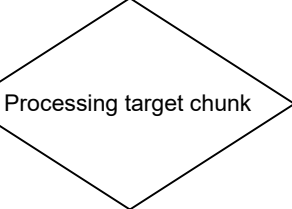
Chunks that match on more SP's are considered better matches than those that match on fewer SP => base chunk



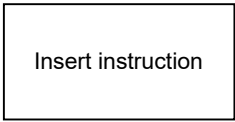
base used for delta compression



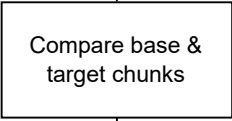
Путем перебора base chunk мы вычисляем хеш-функции в выбранных позициях, сохраняем хеш-функции и смещения в index.



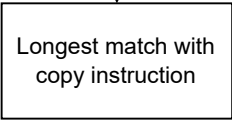
Вычисляем хеш-функции в позициях скользящего окна. Ищем соответствие хеш-функции в индексе, чтобы найти совпадение с базовым фрагментом.



Если байты не совпадают, мы выполняем инструкцию вставки, чтобы вставить байты target фрагмента в буфер вывода. Также добавляем эту область в индекс хешей



Если есть совпадение, мы сравниваем байты в base и target фрагментах вперед и назад от начальной позиции, чтобы создать максимальное возможное совпадение.



исляем
циях,
е в temp

циях

ий в
ие с

В процессе обратных
проходов мы можем
столкнуться с регионом,
который был ранее
закодирован.

Мы решаем, следует ли
сохранить предыдущую
инструкцию или обновить ее
для достижения лучшей
степени сжатия.

Save/Update
previously encoded

