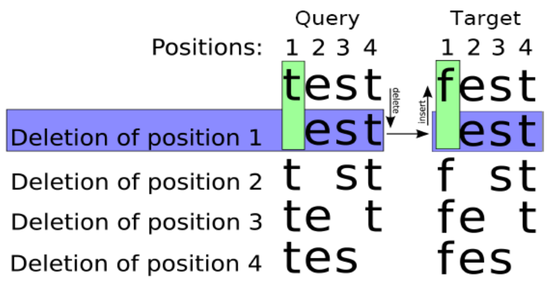
FastSS: Using a Deletion Neighbourhood for Faster Approximate Keyword Searching

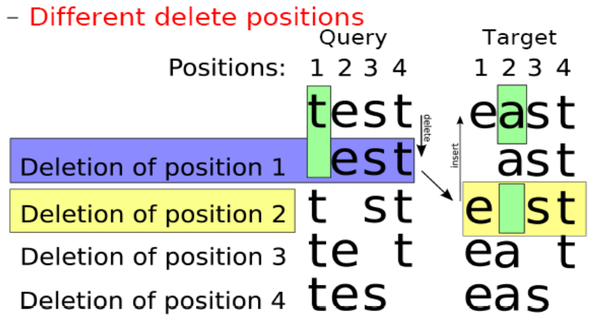
Thomas Bocek[1], Ela Hunt[2] and Burkhard Stiller[1,2]
[1]University of Zurich and [2]ETH Zurich

RESULTS:
2. Definition of a deletion neighbourhood
3. Derivation of edit distance via deletion neighbourhood
4. FastSS in three variants: tradeoff between space and time
5. Experimental comparison: fastSS is faster than a linear scan, a trie, n-gram and all other known methods

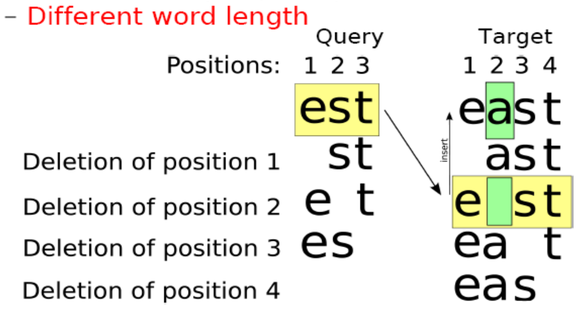
Edit distance (test,fest)=1



Edit distance (test,east) =2
– Different delete positions



Edit distance (est,east) = 1
– Different word length

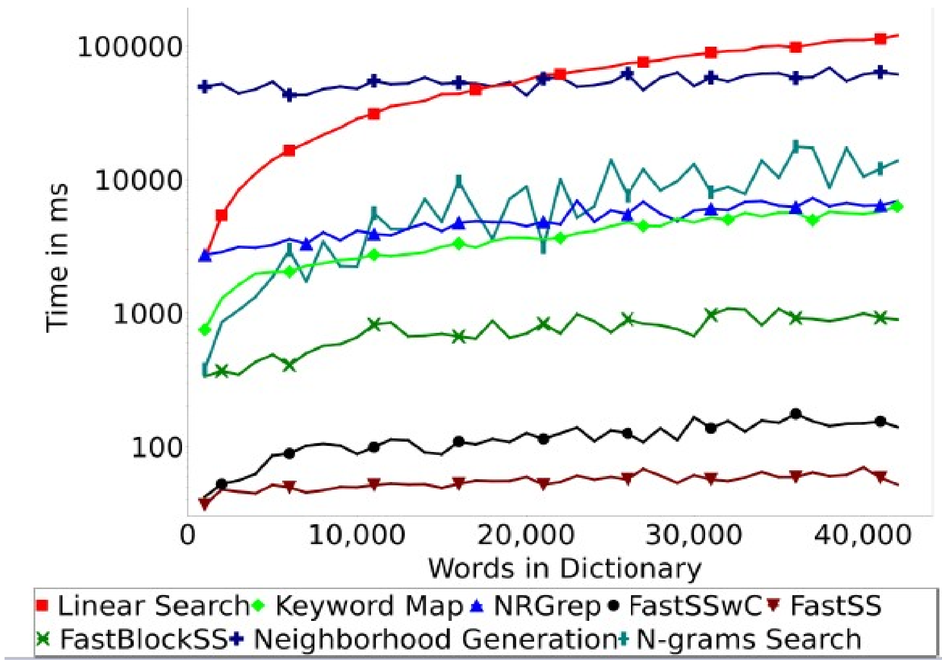


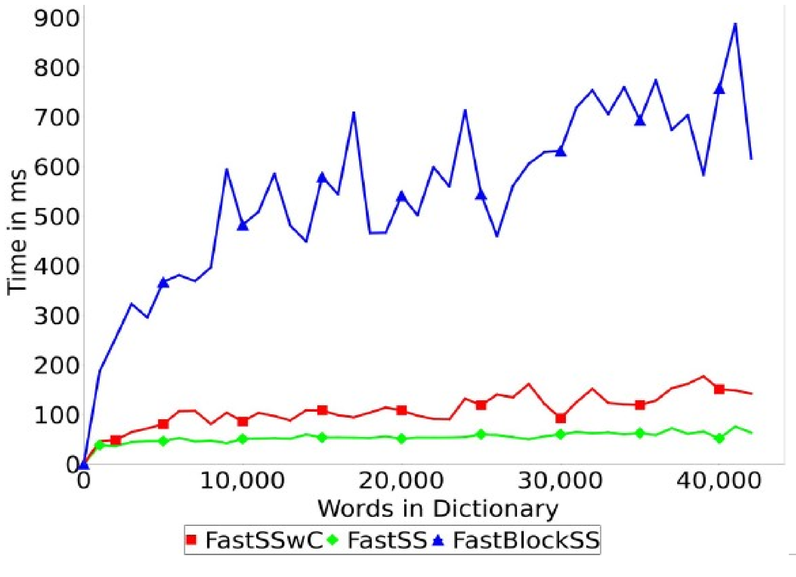
Deletion neighbourhood:

Ud (word, k) = list of all words with up to k deletions (optionally with lists of deletion postions)

Ud can model edit distance:

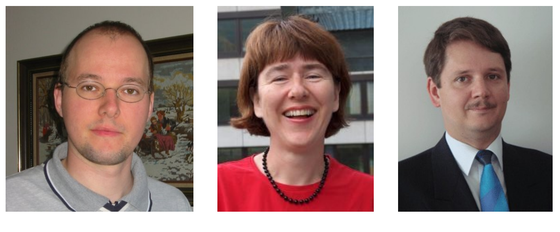
$$f(x,y) = \begin{cases} 0 & : & x = [], y = [] & (1) \\ |x| & : & x \neq [], y = [] & (2) \\ |y| & : & x = [], y \neq [] & (3) \\ 1 + f(tl(x), y) & : & hd(x) < hd(y) & (4) \\ 1 + f(x, tl(y)) & : & hd(x) > hd(y) & (5) \\ 1 + f(tl(x), tl(y)) & : & hd(x) = hd(y) & (6) \end{cases}$$





FastSS

http://fastss.csg.uzh.ch/



Thomas Bocek | Ela Hunt | Burkhard Stiller