Indexed approximate string matching

This is the problem of finding all the approximate occurrences, in a text *T*[1, *n*], of a pattern *P*[1,*m*], both over an alphabet of size *s*. By "approximate occurrence" I mean that at most *k* "edit operations" need to be done on any text substring to make it match the pattern. The most popular edit operations are insertions, deletions, and substitution of characters [1]. In particular I refer to the indexed variant of the problem [2], where one builds an index on *T* to speed up the searches for arbitrary patterns.

Although there has been progress on this problem, one still finds that either the index is of exponential size (in *k* or *m* or *s*), or the search takes exponential time. See e.g. [3, 4]. I believe this is a fundamental space/time barrier, but as far as I know this has not been proved.

References

1. G. Navarro. A guided tour to approximate string matching. ACM Computing Surveys 33(1):31-88, 2001.
2. G. Navarro, R. Baeza-Yates, E. Sutinen, J. Tarhio. Indexing methods for approximate string matching. IEEE Data Engineering Bulletin 24(4):19-27, 2001.
3. R. Cole, L. Gottlieb, M. Lewenstein. Dictionary matching and indexing with errors and don't cares. Proc. STOC'04, pp 91-100, 2004.
4. M. Maas, J. Nowak. Text indexing with errors. Proc. CPM'05, pp. 21-32, 2005.

Gonzalo Navarro,
University of Chile,
Chile.