Big data: Fast, Small, and Read Only Lookup

Contemporary Machine translation:

- Large parallel corpora
- Fast lookup.
- Low storage usage.
- Low Memory usage

Timetable

- Mock data work
- Real data

Alpha 0.1

- First version, working with real data.

Features:

- Fast construction, low memory usage
- Lookup is done with mmap(), low memory usage
- Looking up a phrase that is missing returns „Key not found“.
- Good hashing – 0 clashes over 63 Million unique entries.

Issues

- No duplicate entries support (as of yet).
- No compression.
- Query result may return part of the subsequent entry at the end.
- Memory usage at construction time is dependant on the size of the hash table.

Some numbers

Data size (en-cs model):

- Compressed 859 MB
- Uncompressed 4.7 GB

Post processing size:

- Hash table – 650 MB
- Data – 3.2 GB
- ~26% compression over uncompressed data.

Time:

- Time 62 seconds, 4900MQ, USB 3.0 5200 RPM HDD

Usage:

- Max memory usage – 660 MB

Some numbers 2

Query times

- Maximum, 300ms
- Minimum, 0 ms
- Depends on hits and misses in the mmap'ed file. No medium really.

Memory usage.

- After looking up 7 phrases, memory usage was about 1.5 MB

Demonstration...

Updated timetable

- December – Fix usability issues, investigate compression
- January – Benchmark various compression techniques, Testing
- February – Start writing a report and add minor enhancements if necessary.
- March – Have report finished and polished, code changes only if absolutely necessary.

Thank you

- Thank you for your time.
- Special thanks to Hieu for pointing out silly mistakes such as „Your index starts from 1“ or „You forgot to recompile“.
- Special thanks to Hieu for pointing out not so silly mistakes such as „You are not using mmap() correctly“ and „You are doing it wrong!“