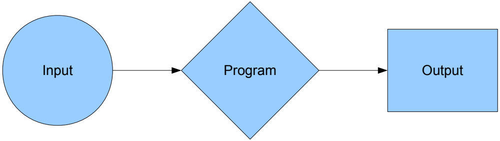
Blessing and Curse of large data

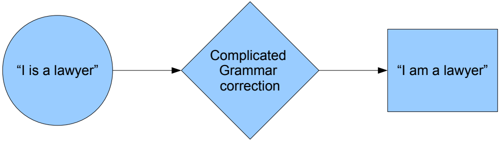
Edo Liberty

Old programming paradigm



- The input is small and the program can store/read it many times
- There is a lot of domain intelligence built into the program

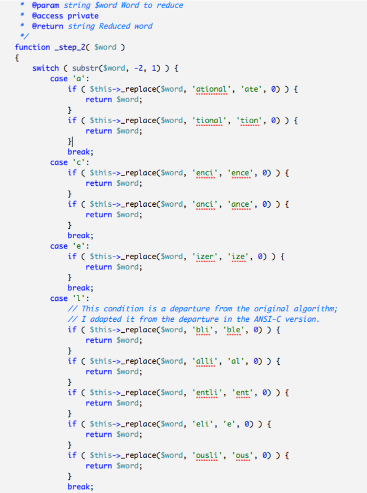
Old programming paradigm



- A short sentence is given to a grammar correction software.
- Programers and linguists produced code which is highly specialized.

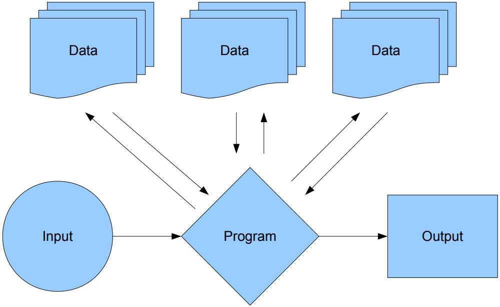
Old programming paradigm

Part of a stemming module (tiny fraction of the whole process)



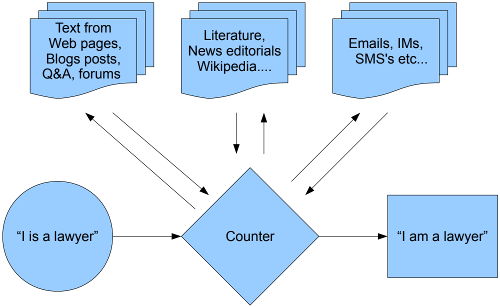
```
 * @param string $word Word to reduce
 * @access private
 * @return string Reduced word
 */
function _step_2( $word )
{
    switch ( substr($word, -2, 1) ) {
        case 'a':
            if ( $this->_replace($word, 'ational', 'ate', 0) ) {
                return $word;
            }
            if ( $this->_replace($word, 'tional', 'tion', 0) ) {
                return $word;
            }
            break;
        case 'c':
            if ( $this->_replace($word, 'enci', 'ence', 0) ) {
                return $word;
            }
            if ( $this->_replace($word, 'anci', 'ance', 0) ) {
                return $word;
            }
            break;
        case 'e':
            if ( $this->_replace($word, 'izer', 'ize', 0) ) {
                return $word;
            }
            break;
        case 'l':
            // This condition is a departure from the original algorithm;
            // I adapted it from the departure in the ANSI-C version.
            if ( $this->_replace($word, 'bli', 'ble', 0) ) {
                return $word;
            }
            if ( $this->_replace($word, 'alli', 'al', 0) ) {
                return $word;
            }
            if ( $this->_replace($word, 'entli', 'ent', 0) ) {
                return $word;
            }
            if ( $this->_replace($word, 'eli', 'e', 0) ) {
                return $word;
            }
            if ( $this->_replace($word, 'ousli', 'ous', 0) ) {
                return $word;
            }
            break;
```

New programming paradigm



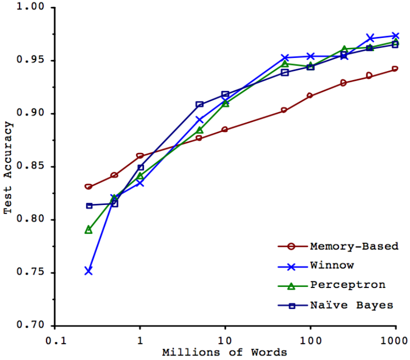
- There is a huge (virtually infinite) amount of data
- The "brain" is the data and not the program

New programming paradigm


Text from Web pages, Blogs posts, Q&A, forums
Literature, News editorials Wikipedia....
Emails, IMs, SMS's etc...
“I is a lawyer”
Counter
“I am a lawyer”

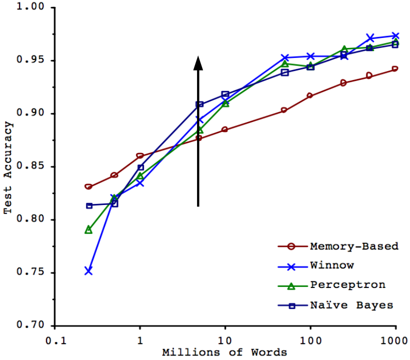
- “**I is a lawyer**” appeared **800,000** times usually like “**i) is a lawyer ...**” or “**George I. is a lawyer**” etc.
- “**I am a lawyer**” appeared as is **1,200,000** in respected sources.

New programming paradigm



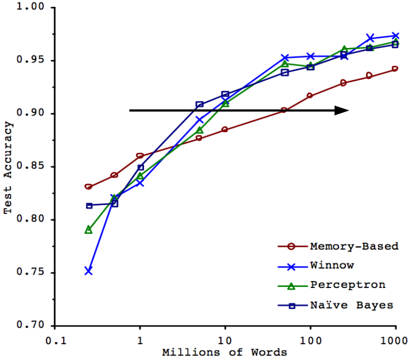
Michele Banko, Eric Brill: Scaling to Very Very Large Corpora for Natural Language Disambiguation.

New programming paradigm



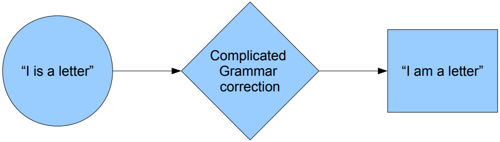
Clearly, some algorithms perform better than others.

New programming paradigm



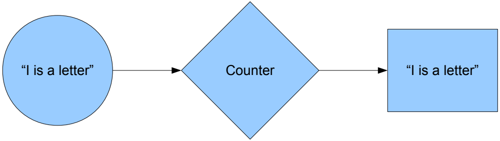
But, having more data is sometime more important than the algorithm...

New programming paradigm



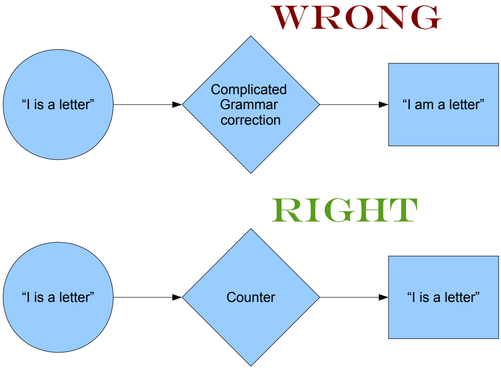
- “**letter**” and “**lawyer**” are both nouns
- “**I is a letter**” corrected to “**I am a letter**”

New programming paradigm



- "**I is a letter**" appeared **5,000,000** times
- "**I am a letter**" appeared only **200,000** times

New programming paradigm



- Could this be done in the old paradigm?
- How about grammar correcting Hungarian now?

Other examples where the data is “everything”

- Ranking / sorting search results
- Web advertising
- Text and image search
- Recommendation engines
- Fighting web abuse (spam, malware etc...)
- Spelling, Suggesting
- many many more...

So, you want more data?

Careful what you wish for!

- King James **Bible** - **1.4 MB**
- Only text on **WIkipedia** - **6.1 GB** (1GB = 1000MB)
- All ***.gov domain** on the web **1 TB** (1TB = 1000GB)
- Incoming **daily emails**[1] to Yahoo! **1-10 PB** (1PB = 1000GB)
- Size of the **internet**[2] **10 Exa-byte** (1EB = 1000 PB)

"Simply counting" doesn't sound so easy anymore....

[1] This number depends on whether you count spam, forwards, attachments and so on. The number I give is a conservative (and intentionally obfuscated) estimate of the amount of new raw text.

[2] The size of the internet is not really known, and it is even unclear what the word "size" means exactly in this context. However, 10EB is, to the best of my knowledge, a conservative estimate of the amount of text in publicly accessible static p

The bread and butter of large data handling

- Massively distributed clusters
  (thousands of machines in each warehouse)
- Software abstraction to cluster
  (recovers from single node crushed etc.)

Still, working with computer clusters is an art in its own right.

- Slow communication (compared to hard drive access)
- Communication is unreliable
- Failures are often and recovery is time consuming
- Programing is complicated (good programing is very complicated).
- Many tasks are simply impossible...

Even for companies like Yahoo!, Google, and Amazon, this is a massive undertaking and a never ending effort.

The bread and butter of large data handling

On the algorithmic side:

- New complexity classed and considerations (communication complexity)
- New computational frameworks (like map-reduce or message passing)
- New algorithms, data structures, and computational models (e.g., search indexes, streaming)

What is the goal of this class?

It turns out the a rather small set of tools gives us many and wonderful abilities. Most of these are randomized in nature:

- Estimating
- Sampling
- Hashing
- Streaming
- Sketching
- Embedding
- and Indexing

The goal of this class to make you as comfortable with these ideas as possible by learning many different manifestations of them.