



Large Corpus Construction for Chinese Lexical Development

John O'Neil & Thomas Emerson
Basis Technology Corporation
{oneil,tree}@basistech.com

March 8, 2006

Basis Technology Corporation

P 617.386.2000
800.697.2062 (toll-free)
F 617.386.2020
W info@basistech.com
www.basistech.com



Problem Definition

- For natural language applications, we need more training data.
- For language data, the World Wide Web is the ultimate fire hose.
- But how do you drink from it?



Overview

- Our motivation: Chinese text segmentation.
- Our goal: Chinese lexical development.
- Our technique: unsupervised machine learning.
- Our means: corpus collection from the WWW.



Chinese Text Segmentation

- Chinese is written without spaces between words.
- Most natural language applications operate on words.
- So, we need to find the word boundaries.



Chinese Word Segmentation Approaches

- Purely lexical.
- Purely statistical.
- Hybrid approaches.

They all need a substantial corpus.



A 鸡和蛋 Problem

- We need tokenization to get a lexicon and lexical statistics.
- We need a lexicon and lexical statistics for the tokenizer.
- Which comes first?

In practice, use a pre-existing "core" lexicon.



Unsupervised Training for Chinese Word Segmentation

- Problem: finding additional lexical information.
 - POS tags.
 - Simplified vs. traditional versions.
- Problem: find a definition for "word" and consistently apply it.
- A better solution: use it to find new words in a document stream.
 - Works continuously.
 - Can be alert for a "burst" signaling a new word.
- Other unsupervised techniques.
 - Clustering words and contexts for classification or POS assignment.
 - Clustering documents and assigning topics.



Overview: Building a Large Chinese Corpus

- Crawling
- Post-Processing



Crawler Desiderata

- Juggling $\gg 10^6$ URIs.
- Minds its manners.
 - Obeys robot-exclusion preferences.
 - Doesn't thrash sites.
- Directed downloading.
 - We're only interested in text – Chinese text, in fact.
 - Everything else is wasted bandwidth and storage.



Crawler Desiderata (2)

- Access to the data while the crawl continues.
 - Since we don't want to wait for the crawl to end.
 - Since the goal is to crawl continuously...
- Recrawling rapidly changing or dynamic sites.
 - News
 - Blogs



Heritrix

- Fits our needs well.
 - Handles the URIs.
 - Is polite.
 - Can be told what to ignore.
 - Stores data in 100MB ARC files.
- Other qualities:
 - Good UI for monitoring crawls in progress.
 - Open-source and active development community.

<http://crawler.archive.org/>



Other Crawling Issues

- Seed generation
 - Randomly selected from the Open Directory Project.
 - Under “Top:World:Simplified Chinese”
- Job Configuration
 - Filtering downloads based on 176 filename suffixes.
 - “default-encoding”
 - “Accept-Language”
- Handling the data
 - ARC files
 - Disk contention issues



Crawl Setup

- Start with 1,500 randomly selected ODP URIs.
- Heritrix 1.4-pre
- Sun JDK 1.4.2 (512 MB heap)
- 666MHz dual CPU, 1GB RAM
- Started with 50 threads, increased to 150.



First Crawl Experience

- Ran 11 days.
- Stopped when disk space exhausted.
- Over that time, stored 7,372,351 URIs.
 - ~27,926 per hour.
 - ~8 documents per second.



Statistics on the First Large Chinese Crawl

URIs stored: 7,372,351

ARC files: 300

Total ARC File Size: 28 GB

Unique Hosts Crawled: 4032

Total HTML size: 109.7 GB

Total Stripped size: 15.8 GB

Languages found: 28



Languages Found on First Large Chinese Crawl

Simplified Chinese	5,510,748	Romanian	52
Traditional Chinese	50,030	Persian	38
Russian	5,986	Hungarian	32
Japanese	4,059	Finnish	28
Korean	393	Bulgarian	26
Arabic	365	Spanish	11
Polish	198	Albanian	11
Greek	136	Vietnamese	10
Thai	120	Swedish	8
Turkish	83	Latvian	5
Czech	67	German	5
Portuguese	65	Icelandic	3
Hebrew	58	Slovak	2
Lithuanian	55	French	1



Disk Issues

- Biggest factor on length of crawl.
- Most disk space used for keeping state than for ARC files.
 - 48GB vs. 28 GB.
 - State was large proportion because of selective downloading.
- Disk contention became a performance bottleneck.
 - Multiple ARC file writers.
 - Log files.
 - State data.
 - These can be split across disks.



Post-Processing

- Extract interesting documents from ARC files.
- Transcode into UTF-8.
 - Expand HTML character entities (中)
- Strip HTML (and check that file size > 1024 bytes).
 - Aware of broken/non-standard markup.
 - ...but try to maintain text structure.
 - We used Vilistextum.
- Detect language and encoding.
- Store resulting text files in directory structure.



Post-Processing We Didn't Do

- Duplicate document detection.
- Boiler plate text removal.



Most Statistics on Chinese Crawl

Number of files: 3,291,985
Average file size: 4,935 bytes
Total hanzi: 3,861,758,249



Data Management

- Crawl Data
 - 28 GB compressed.
 - Backups are difficult. (7 DVDs!)
 - Easier with one or more external FireWire HDs.
- Processed Data
 - Each URI's text data stored as a single text file.
 - Millions of files.
 - Therefore, used bzip2 compressed tarfile archives.



Finding Unknown Words in Chinese

- The first use for all this data.
- The goal: to find words in Chinese that aren't in our lexicon.
 - Neologisms.
 - New personal names (e.g. celebrities or newsmakers).
 - New foreign words or names borrowed into Chinese.



Word Finding Algorithm

- An on-line algorithm
 - Adapted from the batch algorithm of Jin et. al.
 - Intended to sit on a constant document stream.
- Proceeds incrementally -- with each new document:
 - The internal data is updated.
 - The document is tokenized.
 - One or more new words might be proposed.



Algorithm Details

- A map of substring and counts is maintained.
 - Substrings must be of length $2 \leq n \leq 12$.
 - Look for all substrings in a document.
 - If it occurs in two or more sentences, it's added to the map.
- During burn-in, only substrings are collected.
 - Burn-in is variable.
 - Tens of thousands of documents is a minimum.



Algorithm Details (2)

- After burn-in, a document is also tokenized.
 - In each sentence, we find all the substrings and counts.
 - We remove substrings, starting with the lowest-count ones.
 - We leave a substring if removing it would leave a >1 character gap.
 - We continue until no substrings overlap.
- A unique segmentation results.



Algorithm Details (3)

- Tokens not in the lexicon are added to the token count map.
- Look for tokens that reach a threshold count.
- Threshold sensitive to how recently the token was first seen.
 - To recognize a “burst” of a new word.
- If it’s reached a threshold, it’s sent for adjudication.



Word Adjudication

- Native speakers of Chinese decide if the proposed words are "real".
- They also add other information.
 - Part of speech.
 - If it's a person's name, organization name, or location name.
- Most new words are proper nouns or common nouns.



Ruminant Adjudication Tool

Dashboard | Queue | Completed | Logged in as tree

Adjudicate a lexeme from the queue [\[refresh\]](#)

等也	<p>政兼內政部政務部長何炳基 等也 蔣總統</p> <p>李必賢 等也 紛紛表示支持他的主張</p> <p>尾燈和營業小客車頂燈 等也 都以條文明文規定</p> <p>陳光復 等也 先後登上宣傳車高呼</p> <p>民進黨立委洪奇昌 等也 陸續來到國安局</p> <p>民進黨立委洪奇昌 等也 陸續來到國安局</p>
----	---

Please adjudicate this lexeme:

Validity	<input checked="" type="radio"/> VALID <input type="radio"/> INVALID
Parts of Speech	<input type="checkbox"/> Abbreviation <input type="checkbox"/> Adjective <input type="checkbox"/> Adverb <input type="checkbox"/> Common noun <input type="checkbox"/> Construction <input type="checkbox"/> Direction word <input type="checkbox"/> Generic noun <input type="checkbox"/> Noun phrase <input type="checkbox"/> Numeric <input type="checkbox"/> Onomatope <input type="checkbox"/> Other <input type="checkbox"/> Phrase <input type="checkbox"/> Profanity <input type="checkbox"/> Pronoun <input type="checkbox"/> Proper noun <input type="checkbox"/> Temporal noun <input type="checkbox"/> Verb
Decomposition Pattern	2
Reading	
Add a comment	
<input type="button" value="Submit"/> <input type="button" value="Skip"/>	





Word Discovery Results

- About 40% of proposed words are accepted by adjudicators.
- Several thousand so far.
- Sources of new words:
 - Lexical innovations.
 - Proper names.
 - Traditional Chinese characters mixed into Simplified texts.



Conclusion

- How we compiled a large corpus of Simplified Chinese.
- We also collected a similar-sized corpus of Traditional Chinese.
- So far, our biggest concerns have been pragmatic:
 - Getting and adding enough disk space to store the data
 - Bandwidth issues
 - Post-processing the data
- Heritrix has worked well for us
- To do:
 - Moving to incremental crawling, rather than crawling jobs
 - New languages



The End

<http://www.basistech.com/knowledge-center/>