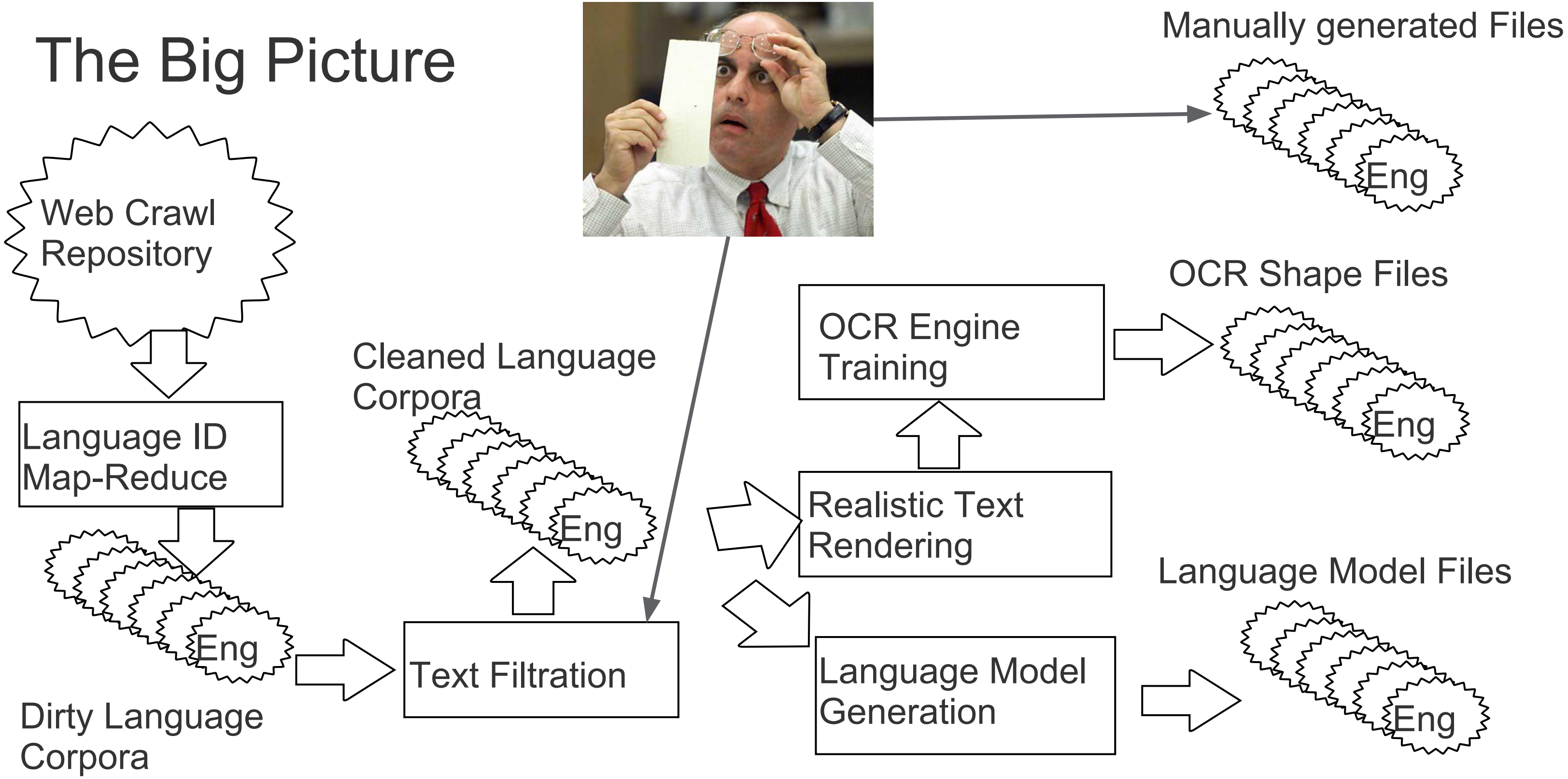
3 Training Tesseract

An Introduction to the Training Process

Ray Smith, Google Inc.

The Big Picture



Web Crawl Repository

Language ID Map-Reduce

Eng

Dirty Language Corpora

Text Filtration

Cleaned Language Corpora

Eng

Realistic Text Rendering

OCR Engine Training

OCR Shape Files

Eng

Language Model Generation

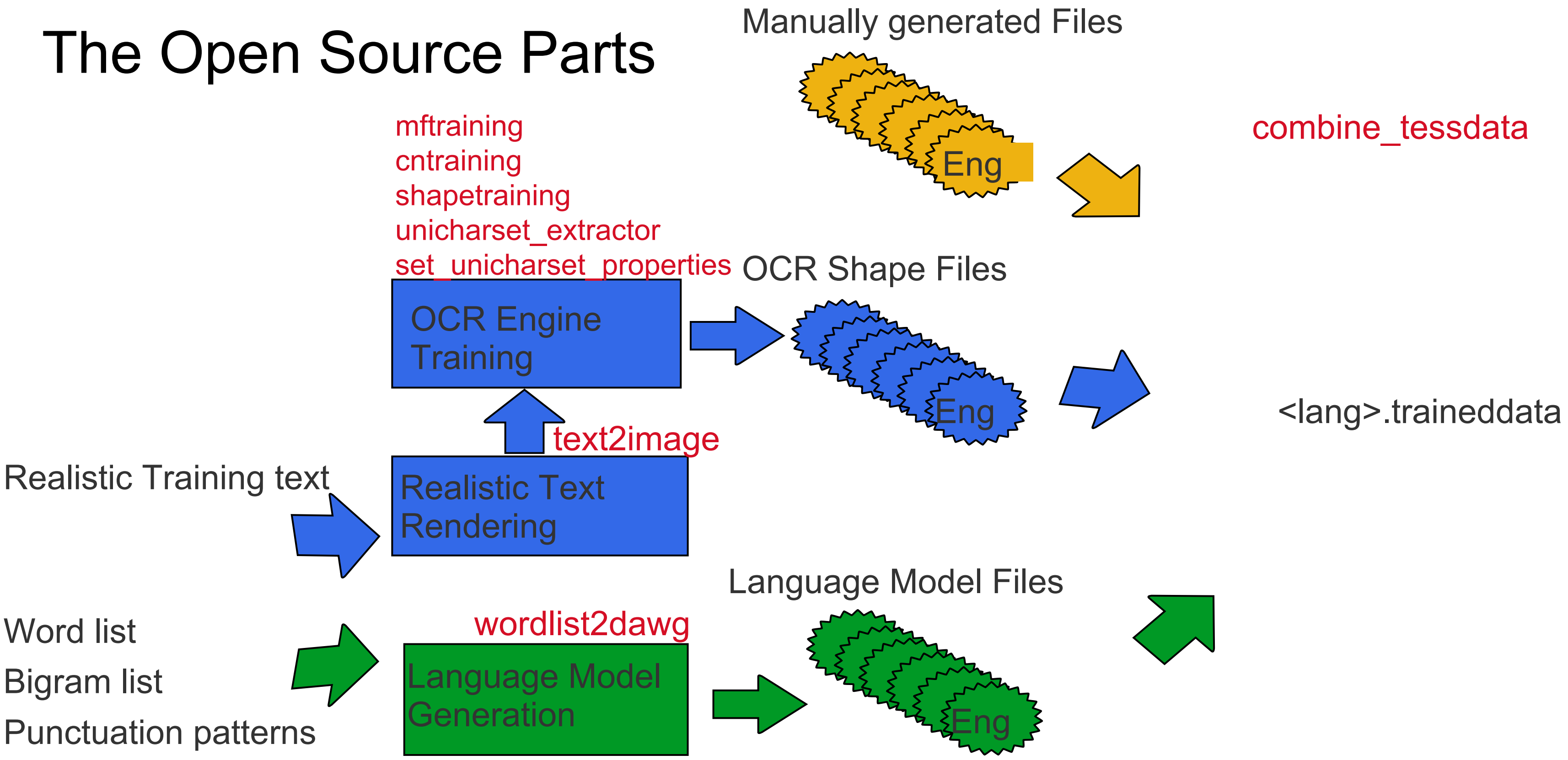
Language Model Files

Eng

Manually generated Files

Eng

The Open Source Parts



Manually generated Files

combine_tessdata

mftraining
cntraining
shapetraining
unicharset_extractor
set_unicharset_properties

OCR Shape Files

OCR Engine Training

<lang>.traineddata

text2image

Realistic Training text

Realistic Text Rendering

Language Model Files

wordlist2dawg

Word list
Bigram list
Punctuation patterns

Language Model Generation

Training Fundamentals

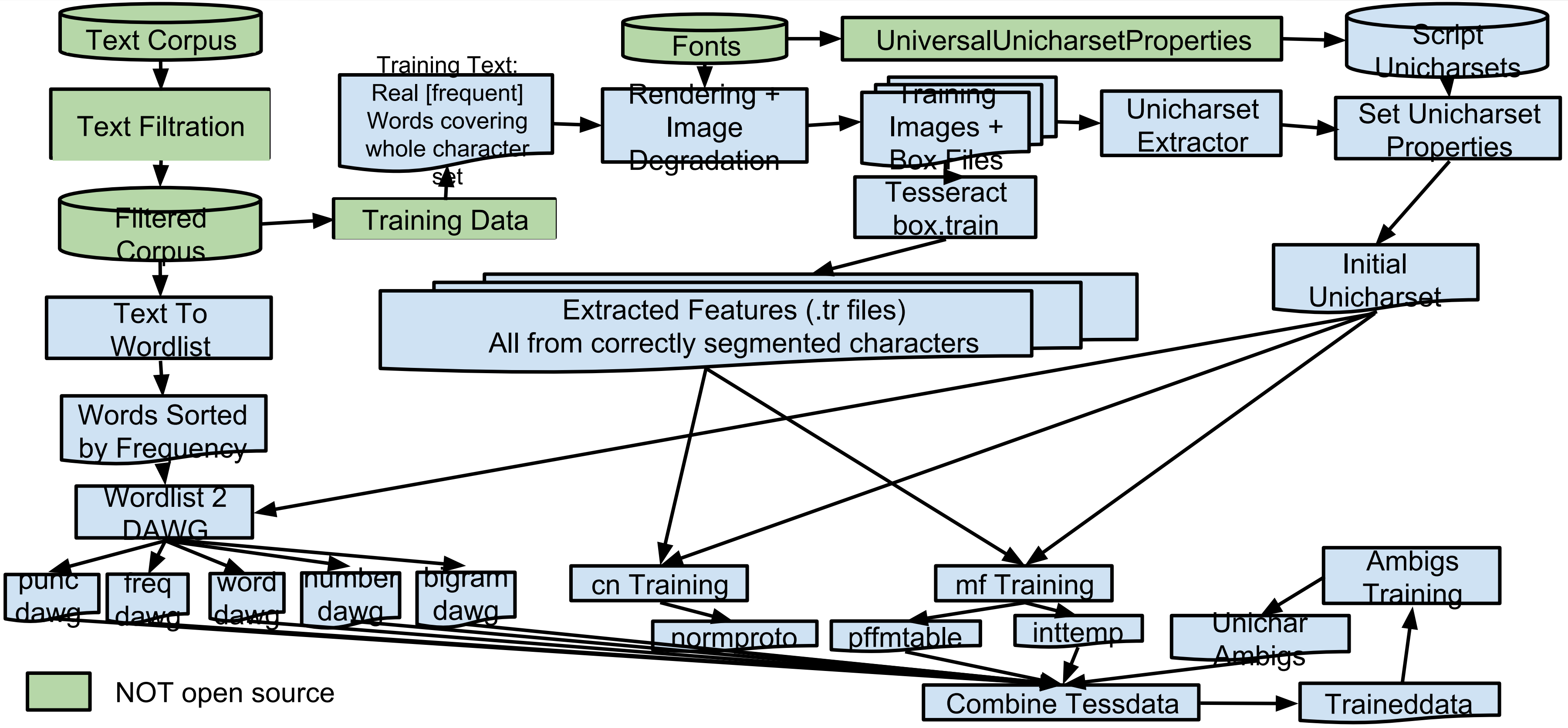
- Character samples **must** be segregated by font
  => Trained on synthetic (rendered, distorted) data
- Few samples required (4-10 of each combination is good. 1 is OK)
- Not many fonts required. (32 used for Latin)
- Not many fonts allowed. (MAX_NUM_CONFIGS=64: Long story.)
- Number of different “characters” now limited only by memory.

What Data Needs to be Created by Training? (1)

Name	Type	Status	Creator	Description
config	Text	Optional	Manual	Lang-specific engine settings if needed
unicharset	Text	Mandatory	unicharset_extractor	The set of recognizable units
unicharambigs	Text	Optional	Manual*	Intrinsic ambiguities for the language
inttemp	Binary	Mandatory	mftraining	Classifier shape data
pffmtable	Text	Mandatory	mftraining	Extra classifier data (num expected features)
normproto	Text	Mandatory	cntraining	Classifier baseline position info
cube-unicharset	Text	Optional	unicharset_extractor	Cube’s set of recognizable units
shapetable	Binary	Optional	shapetraining	Indirection between classifier and unicharset
params-model	Text	Optional	Google tool	Alternative method for combining LM & classifier

Inside tesstrain.sh

Input	Program	Output
Realistic Training Text	text2image	*.tif, *.box
*.box	unicharset_extractor	unicharset
unicharset, <script>.unicharset	set_unicharset_properties	unicharset
Word List	wordlist2dawg	word-dawg
Frequent Word List	wordlist2dawg	freq-dawg
*.tif, *.box	tesseract	*.tr
*.tr	cntraining	normproto
*.tr	mftraining	inttemp, pffmtable
unicharset, dawgs, normproto, inttemp, pffmtable, config	combine_tessdata	traineddata



🟩 NOT open source

Overview of Tesseract Training Process

Language-Specific Data

- Training text: defines the character set.
- Wordlists: define the language model. (Including bigrams when present.)
- Pango layout: defines the grapheme clusters (recognition units).
  Eg: 0xca6 + 0xccd + 0xca6 + 0xcc7 ->
  ದ ದ ೇ ದ್ದೇ
- ICU: determines what is right-to-left.
- Script.unicharset: Stores typical font metrics for unicode chars.
- Config files: in training/langdata/<lang>/<lang>.config.
- Vertical rendering: Determined manually.

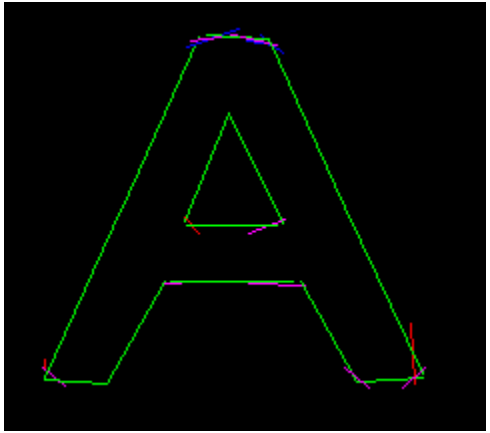
MFtraining

Input: font.tr files (Stored Features with utf-8 labels)
Output: inttemp (knn classifier data)
pffmtable (helper information for class pruner)
Operation:

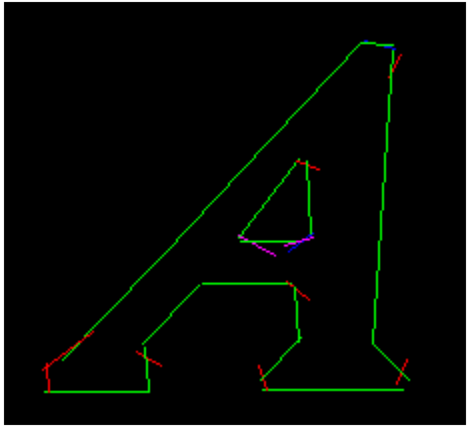
1. Independently cluster features in each font/char class combination.
2. Combine similar cluster means across fonts (single char class).
3. Define each font/char class as a combination of cluster means (a font config).
4. Build class pruner and main knn classifier.

Clustering Result

Protos of Arial ‘A’



Protos of Times Italic ‘A’



CNtraining

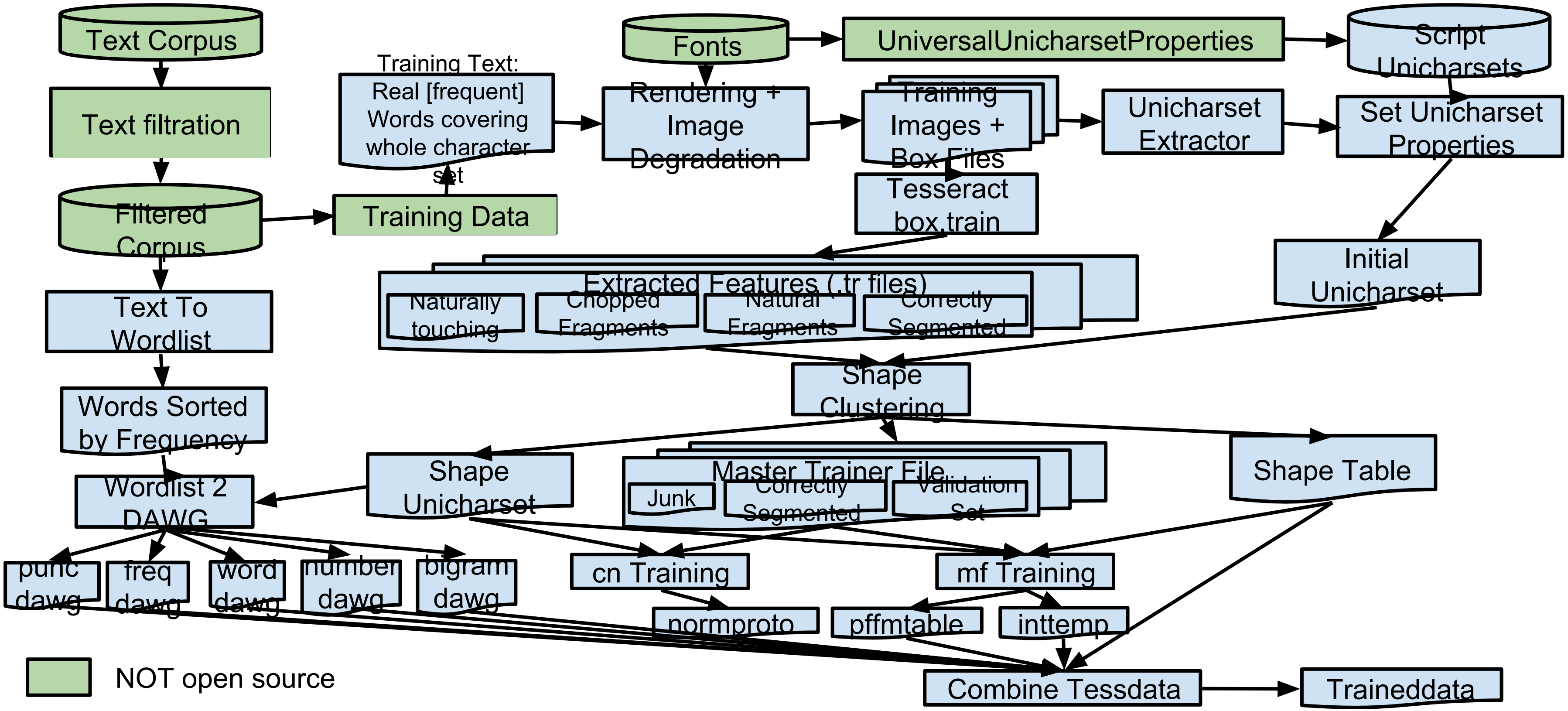
Input: font.tr files (Stored Features with utf-8 labels)
Output: normproto (GMM means of the CN feature)
Operation:

1. Independently cluster the CN feature of all fonts for each char class.
2. Write cluster means (a Gaussian Mixture Model) to normproto file.

Shapetraining

Input: font.tr files (Stored Features with utf-8 labels)
Output: shapetable (Mapping from an index to a collection of unichar-ids, fonts)
Operation:

1. Cluster across all fonts and all char classes.
2. Merge ambiguous classes into shapes.
3. Works OK for Indic, but not so good for others.



Text Corpus

Text filtration

Filtered Corpus

Text To Wordlist

Words Sorted by Frequency

Wordlist 2 DAWG

punc dawg

freq dawg

word dawg

number dawg

bigram dawg

Training Text: Real [frequent] Words covering whole character set

Training Data

Fonts

Rendering + Image Degradation

UniversalUnicharsetProperties

Script Unicharsets

Training Images + Box Files

Unicharset Extractor

Set Unicharset Properties

Tesseract box.train

Initial Unicharset

Extracted Features (.tr files)

Naturally touching

Chopped Fragments

Natural Fragments

Correctly Segmented

Shape Clustering

Shape Unicharset

Master Trainer File

Junk

Correctly Segmented

Validation Set

Shape Table

cn Training

mf Training

normproto

pffmtable

inttemp

Combine Tessdata

Traineddata

🟩 NOT open source

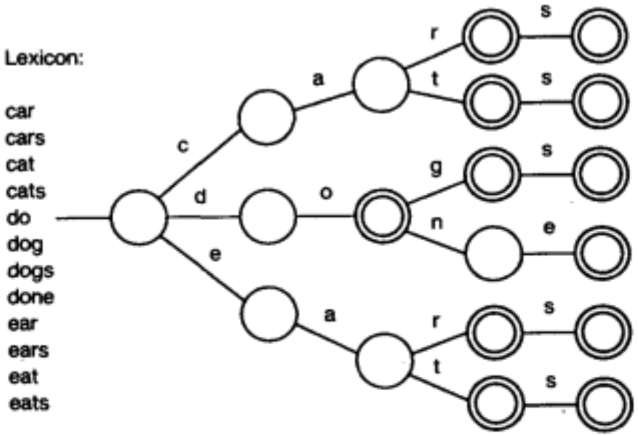
Overview of Tesseract Training Process with Shapes

DAWGs (Directed Acyclic Word Graph)

From “The world’s fastest Scrabble program” A. W. Appel, G.J. Jacobson, *CACM* **31**(5) May 1988, pp572-585:

“The lexicon represented as a raw word list takes about 780 Kbytes, while our dawg can be represented in 175 Kbytes. The relatively small size of this data structure allows us to keep it entirely in core, even on a fairly modest computer.”

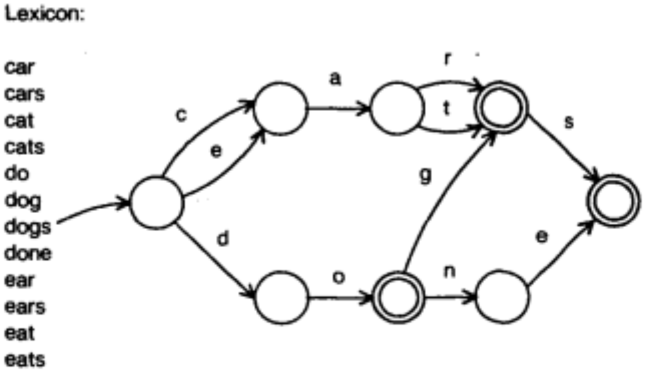
Trie:



Lexicon:

car
cars
cat
cats
do
dog
dogs
done
ear
ears
eat
eats

Dawg:



Lexicon:

car
cars
cat
cats
do
dog
dogs
done
ear
ears
eat
eats

What Data Needs to be Created by Training? (2)

Name	Type	Status	Creator	Description
punc-dawg	Binary	Optional	wordlist2dawg	Patterns of punctuation around words
word-dawg	Binary	Optional	wordlist2dawg	Main word-list/dictionary language model
number-dawg	Binary	Optional	wordlist2dawg	Acceptable number patterns (with units?)
freq-dawg	Binary	Optional	wordlist2dawg	Shorter dictionary of frequent words
fixed-length-dawgs	Binary	Deprecated	wordlist2dawg	Was used for CJK
cube-word-dawg	Binary	Optional	wordlist2dawg	Main word-list/dictionary language model for cube
bigram-dawg	Binary	Optional	wordlist2dawg	Word bigram language model
unambig-dawg	Binary	Optional	wordlist2dawg	List of unambiguous words (not used?)

Thanks for Listening!

Questions?