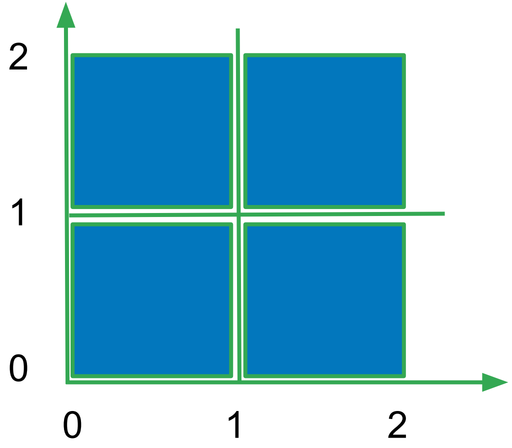
2. Architecture and Data Structures

A quick tour of the Tesseract Code

Ray Smith, Google Inc.

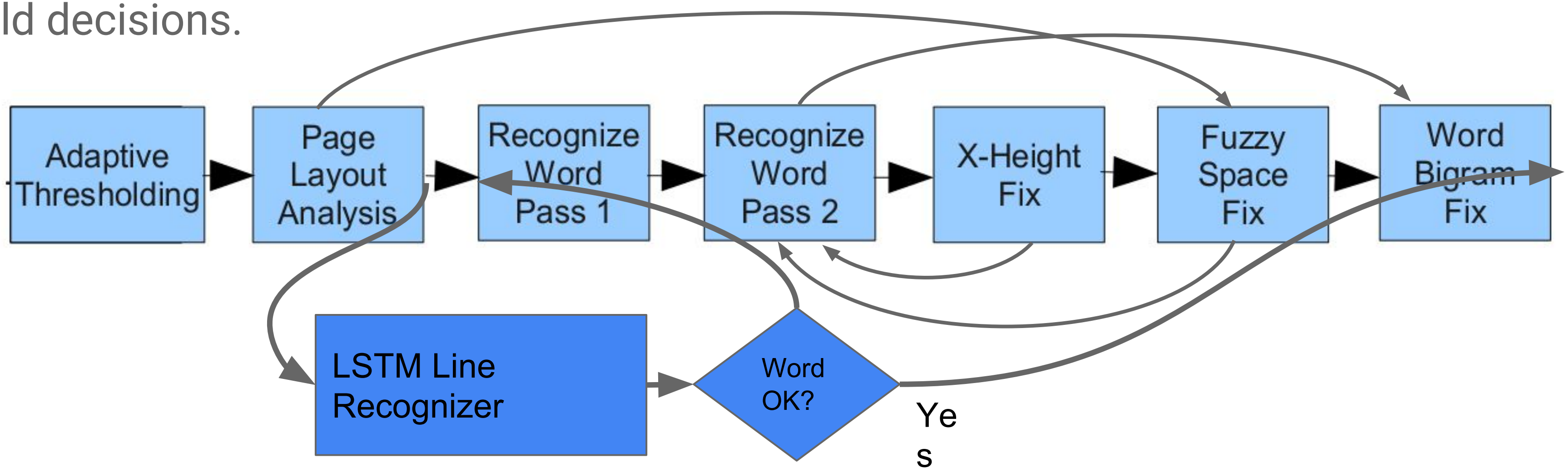
A Note about the Coordinate System

- The pixel edges are aligned with integer coordinates.
- (0, 0) is at **bottom-left.**
- Width = right - left => no silly +1/-1.

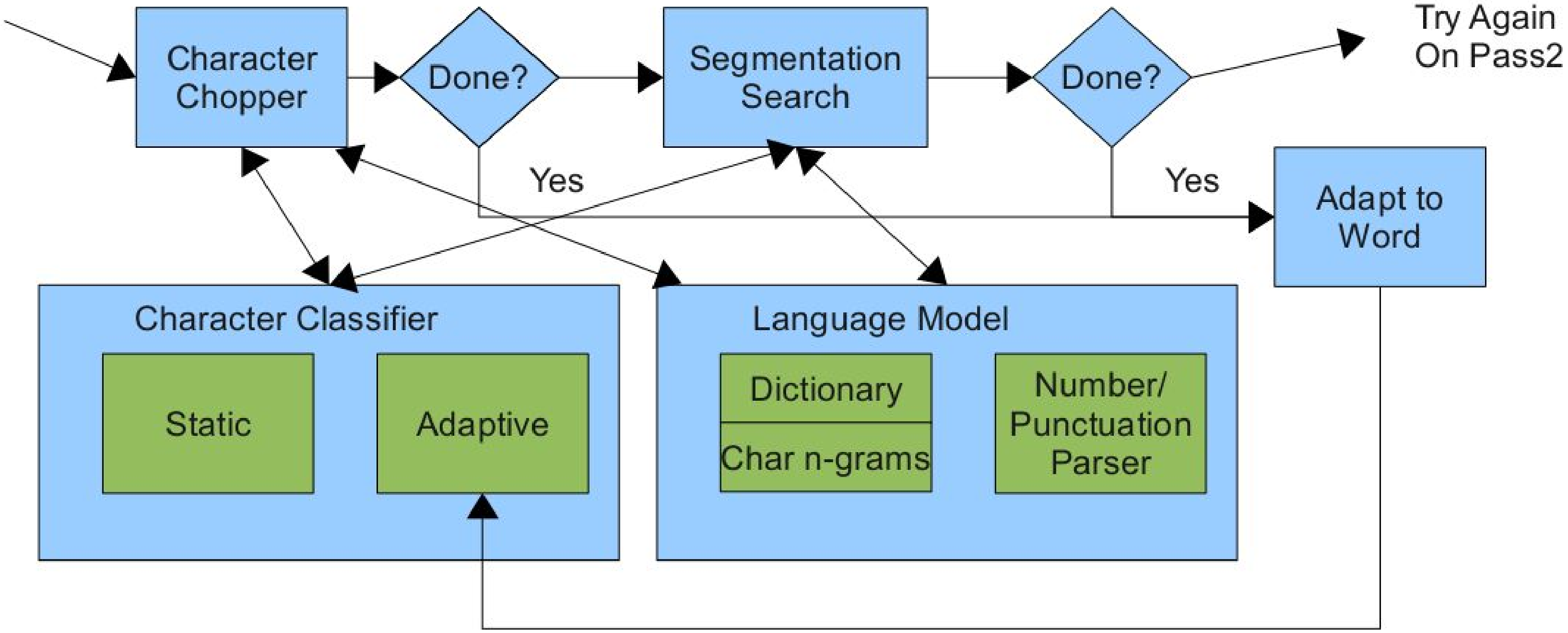


Tesseract System Architecture

Nominally a pipeline, but not really, as there is a lot of re-visiting of old decisions.



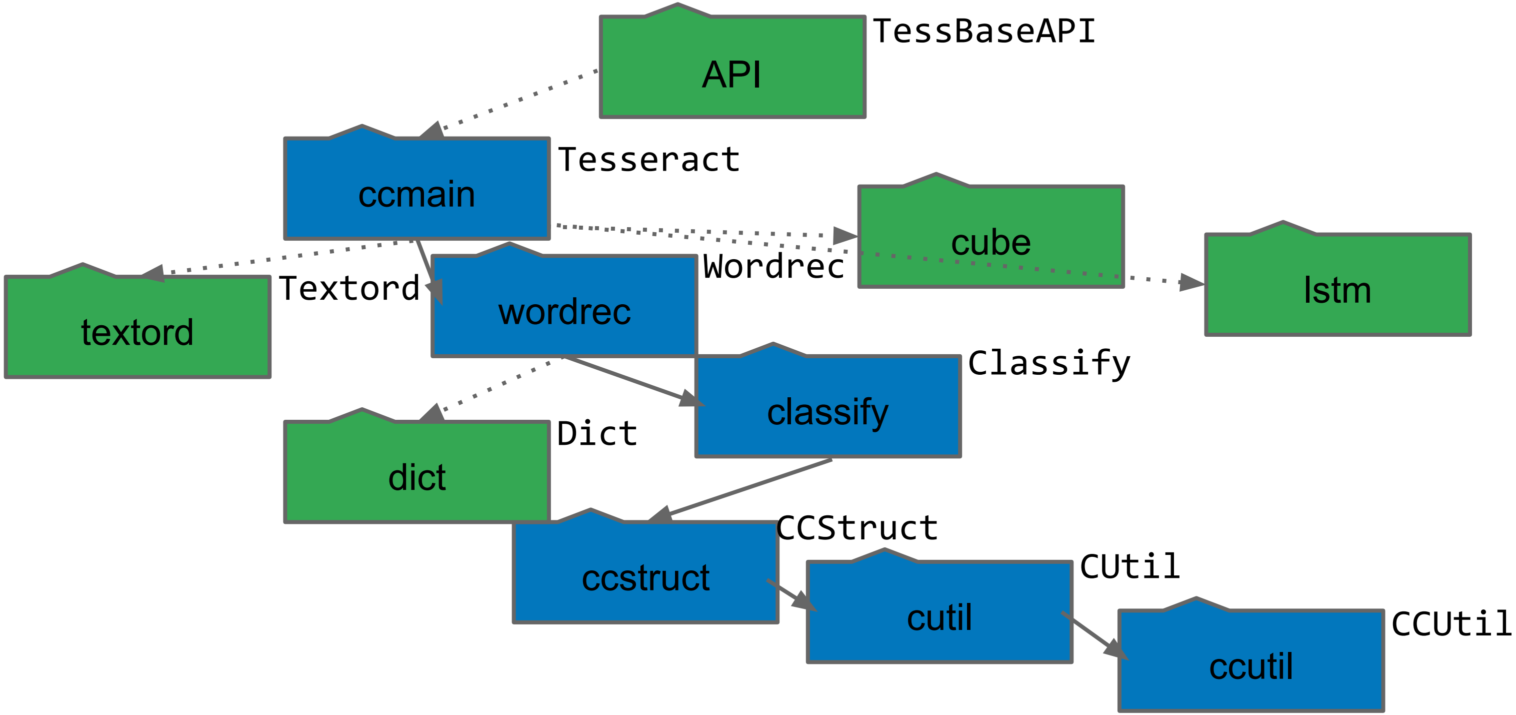
Tesseract Word Recognizer



The ‘C’ Legacy

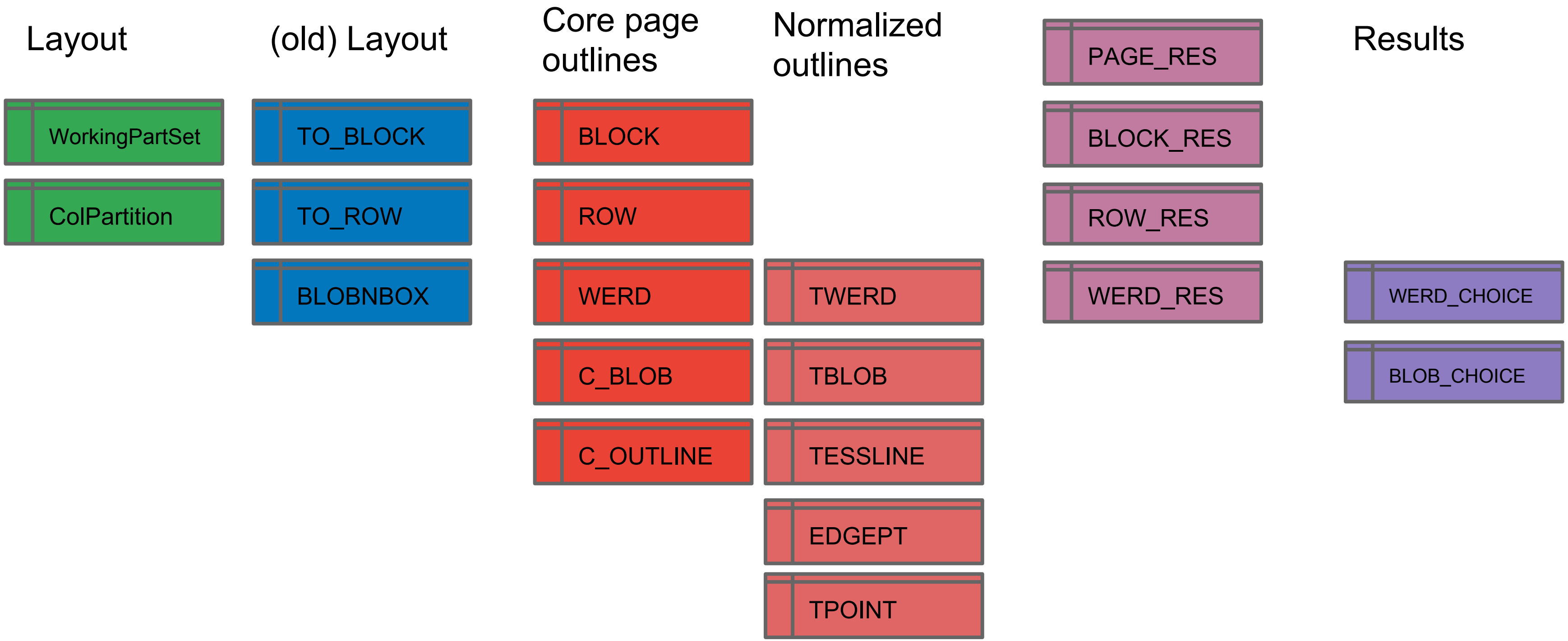
- Large chunks of the code written originally in C.
- Major rewrite in ~1991 with new C++ code.
- C->C++ migration gradual over time since.
- Majority of global functions now live in a convenience directory structure class. (For thread compatibility purposes.)

Directory Structure ~ Functional Architecture



Google

Key Data Structures = Page Hierarchy



Layout

WorkingPartSet

ColPartition

(old) Layout

TO_BLOCK

TO_ROW

BLOBNBOX

Core page outlines

BLOCK

ROW

WERD

C_BLOB

C_OUTLINE

Normalized outlines

TWERD

TBLOB

TESSLINE

EDGEPT

TPOINT

PAGE_RES

BLOCK_RES

ROW_RES

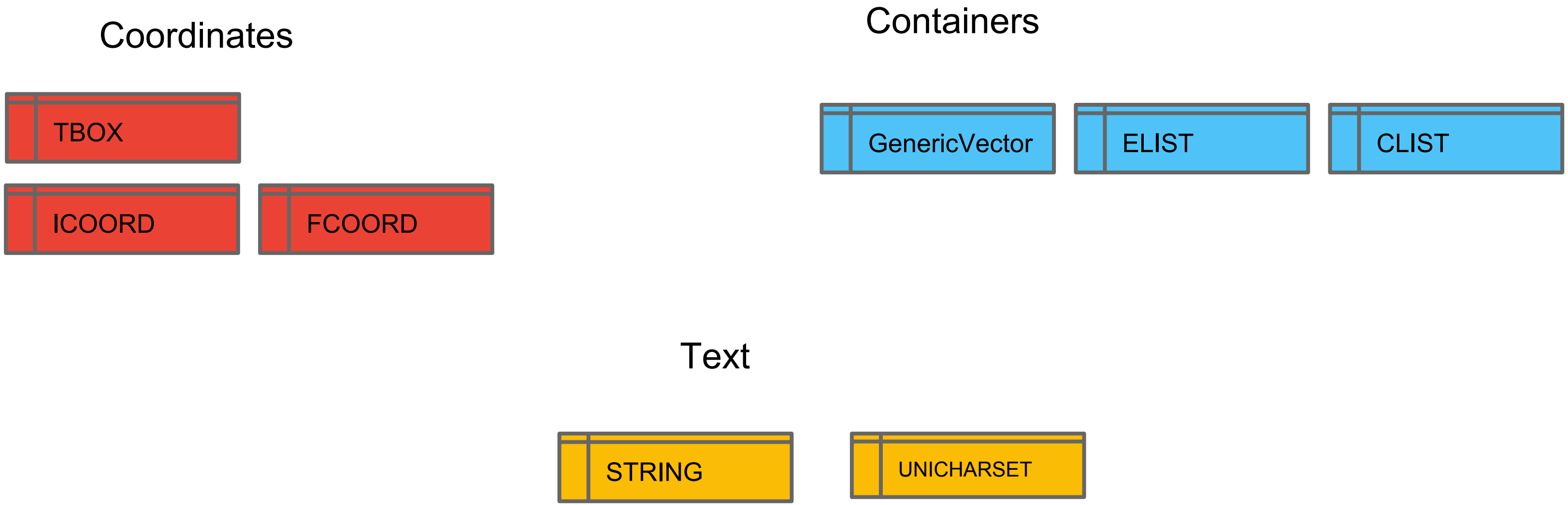
WERD_RES

Results

WERD_CHOICE

BLOB_CHOICE

Software Engineering - Building Blocks



Coordinates

TBOX

ICOORD

FCOORD

Containers

GenericVector

ELIST

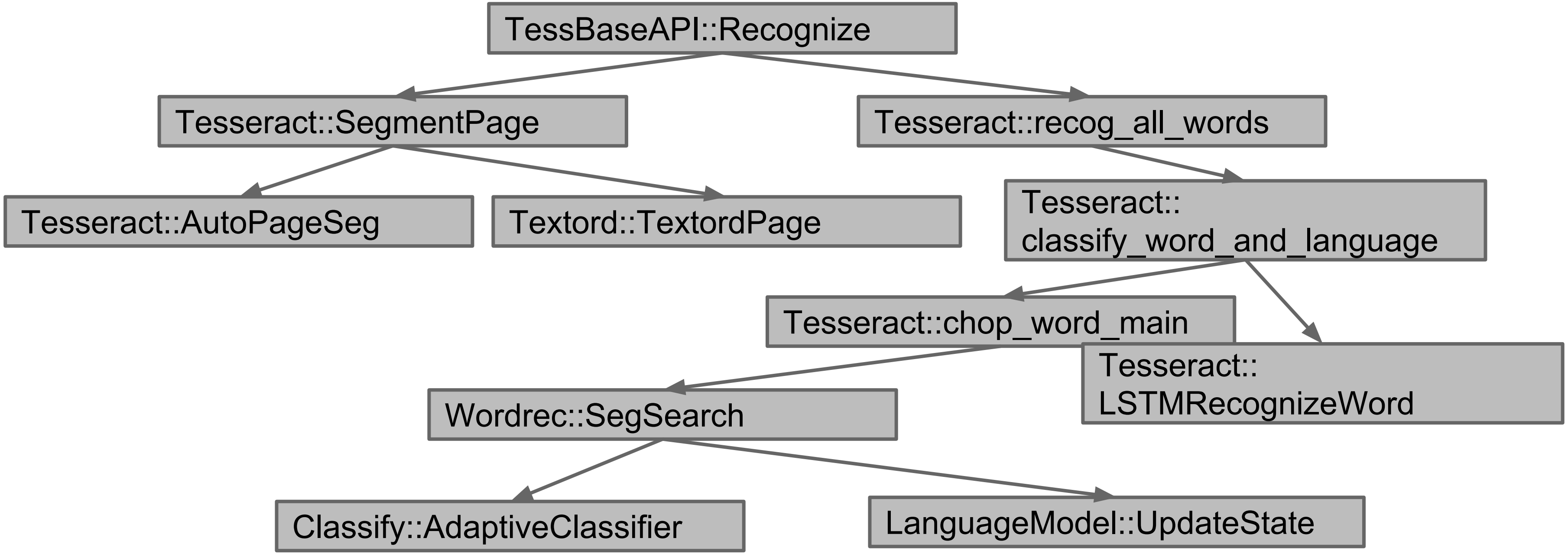
CLIST

Text

STRING

UNICHARSET

Key Parts of the Call Hierarchy



TessBaseAPI::Recognize

Tesseract::SegmentPage

Tesseract::recog_all_words

Tesseract::AutoPageSeg

Textord::TextordPage

Tesseract::
classify_word_and_language

Tesseract::chop_word_main

Tesseract::
LSTMRecognizeWord

Wordrec::SegSearch

Classify::AdaptiveClassifier

LanguageModel::UpdateState

Tesseract’s List Implementation

- Predates STL
- Allows control over ownership of list elements
- Uses macros instead of templates

List Example

ccstruct/blobbox.h:

```
class BLOBNBOX : public ELIST_LINK {
…
};
// Defines classes:
// BLOBNBOX_LIST: a list of BLOBNBOX
// BLOBNBOX_IT:   list iterator
ELISTIZEH(BLOBNBOX)
```

ccstruct/blobbox.h:

```
// Implementation of some of the
// list functions.
ELISTIZE(BLOBNBOX)
```

ccstruct/blobbox.h:

```
float Textord::filter_noise_blobs(
    BLOBNBOX_LIST *src_list,            // original list
    BLOBNBOX_LIST *noise_list,          // noise list
    BLOBNBOX_LIST *small_list) {        // small blobs
  BLOBNBOX_IT src_it(src_list);         // iterators
  BLOBNBOX_IT noise_it(noise_list);
  BLOBNBOX_IT small_it(small_list);
  for (src_it.mark_cycle_pt(); !src_it.cycled_list();
       src_it.forward()) {
    blob = src_it.data();
    if (blob->bounding_box().height() < textord_max_noise_size)
      noise_it.add_after_then_move(src_it.extract());
    else if (blob->enclosed_area() >=
        blob->bounding_box().area() * textord_noise_area_ratio)
      small_it.add_after_then_move(src_it.extract());
  }
```

TessBaseAPI : Simple example

Main API class provides initialization, image input, text/hOCR/PDF output:

```
TessBaseAPI api;
api.Init(NULL, “eng”);
Pix* pix = pixRead(“phototest.tif”);
api.SetImage(pix);
char* text = api.GetUTF8Text();
printf(“%s\n”, text);
delete [] text;
pixDestroy(&pix);
```

TessBaseAPI : Multipage example

```
TessBaseAPI api;
api.Init(NULL, “eng”);
tesseract::TessResultRenderer* renderer =
  new tesseract::TessPDFRenderer(api.GetDatapath());
api.ProcessPages(filename, NULL, 0, renderer);
const char* data;
inT32 data_len;
if (renderer->GetOutput(&data, &data_len)) {
  fwrite(data, 1, data_len, fout);
  fclose(fout);
}
```

ResultIterator for getting the real details

```
ResultIterator* it = api.GetIterator();
do {
  int left, top, right, bottom;
  if (it->BoundingBox(RIL_WORD, &left, &top, &right, &bottom)) {
    char* text = it->GetUTF8Text(RIL_WORD);
    printf("%s %d %d %d %d\n", text, left, top, right, bottom);
    delete [] text;
  }
} while (it->Next(RIL_WORD));
delete it;
```

Thanks for Listening!

Questions?