```
#include <poppler.h>
#include <stdio.h>
#include <string.h>

void usage ()
{
    fputs ("Usage: popplertest <somefile.pdf>\n", stderr);
}

int main (int argc, char** argv)
{
    if (argc != 2) {
        usage();
        return 1;
    }

    g_type_init ();

    gchar *fname;
    char *farg = *(argv + 1);
    gchar *thisdir = g_get_current_dir ();

    if (0 == strncmp ("file://", farg, 7)) {
        fname = g_strdup (fname);
    }
    else if (*farg == ’/’ || *farg == ’~’) {
        fname = g_strdup_printf ("file://%s", farg);
    }
    else {
        fname = g_strdup_printf ("file://%s/%s", thisdir, farg);
    }

    g_free (thisdir);

    GError *err = NULL;
    PopplerDocument *doc =
        poppler_document_new_from_file (fname, NULL, &err);
    if (err) {
        fprintf (stderr,
                 "Error opening %s. Message: %s\n",
                 fname, err->message);
        g_error_free (err);
        g_free (fname);
        return 1;
    }
    g_free (fname);

    PopplerPage *page = poppler_document_get_page (doc, 0);
    gchar* text = poppler_page_get_text (page);

    PopplerRectangle *rects;
    guint n_rects;
    if (!poppler_page_get_text_layout (page, &rects, &n_rects)) {
        fprintf (stderr, "Error: No text.\n");
        g_object_unref (page);
        g_object_unref (doc);
        return 1;
    }

    printf ("String length: %u\nRectangles:    %u\n\n",
            g_utf8_strlen (text, -1), n_rects);

    guint k;
    gchar *pc = text;

    for (k = 0; k < n_rects; k++) {
        PopplerRectangle *pr = rects + k;
        printf ("%c | #<RECTANGLE (%10.6f %10.6f) (%10.6f %10.6f)>\n",
                *(text + k),
                pr->x1, pr->y1, pr->x2, pr->y2);
        pc = g_utf8_next_char (pc);
    }

    for (k = 0; k < n_rects; k++) {
        poppler_rectangle_free (rects + k);
    }
    g_free (rects);

    g_object_unref (page);
    g_object_unref (doc);

    return 0;
}
```