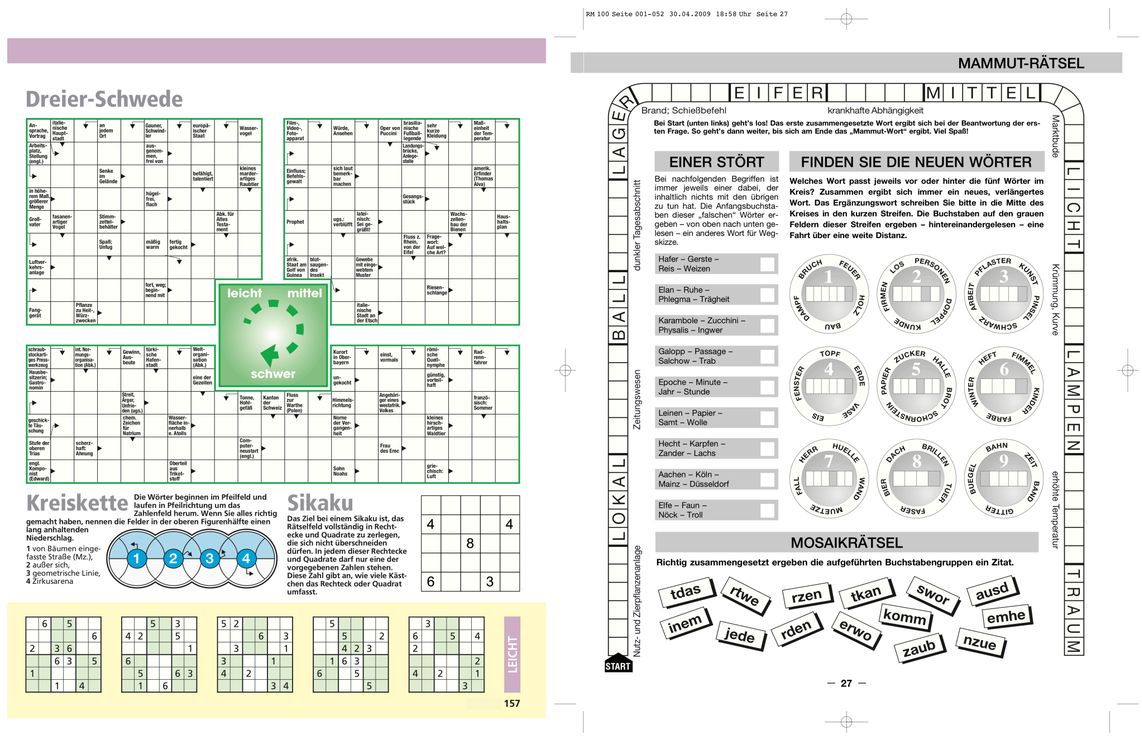
Duplicate Detection for Puzzles in Puzzle Documents

Oskar Bechtold
Institute of Interactive Systems and Data Science

Problem

Rätsel Krüger creates and sells puzzles of different types. Puzzles can be reused and sold multiple times. To prevent selling a puzzle multiple times to the same customer or publishing it too often in a short period of time a system is needed to organize the puzzles. In this work, a near duplicate search was developed and evaluated to add puzzle docuements to duplicate groups. Technology used is Drupal (PHP), Solr (Java) and custom scripts (node.js and python).

- Documents are pdf files
- Each document holds a page for a magazine etc.
- Documents hold one one or puzzle
- Puzzle is an abstract concept
- Files containing the same puzzle should be identified
- Near duplicate detection based on Solr MoreLikeThis



Two example puzzle files. The left containing a variation of Swedish puzzle, circle chain, Sikaku and Sudoku. The right contains mosaic puzzle, false detection, word chains and word carousel.

Puzzle Types

Focus for this research was set on text based puzzles to work with text analysis. A random selection of available puzzle types was chosen to find out if the puzzle type or the length of a puzzle has an impact on the results. Not considered in this work were any types of image or number based puzzles since these would need additional technologies like computer vision or placement of numbers in relation to each other.

Puzzle Types Evaluated

Blinder Passagier	Schwede Bildwitz	Logical
Endlos Rätsel	Schwere Herz	Mosaik
Karussell	Schwede Teekesselchen	Navigator
Kreiskette	Schwede Zahlen	Wortkarussell
Länder Reise Schwede	Silbentreppe	Wörtersuchen

Objective

Given a puzzle document the system should return a list of other documents that contain the same puzzles. If possible these documents should automatically be connected to the detected puzzle to create duplicate groups. If automation is not possible, manually accepting proposed duplicate groups is an alternative solution.

Software

ERPAL® (Drupal) → RK - DMS (Drupal) → Solr Lucene

Technology Stack

Documents are uploaded via a customized ERPAL (Open Source Drupal ERP System) which organizes Customers and Projects. Files are then stored as Drupal entities in the Document Management System (DMS, based on Drupal) which is used to organize the puzzle files. Solr is used to index the files stored in the DMS and uses Drupal fields as documents fields. To find puzzle duplicates this work evaluates the MoreLikeThis (MLT) search component built into solr.

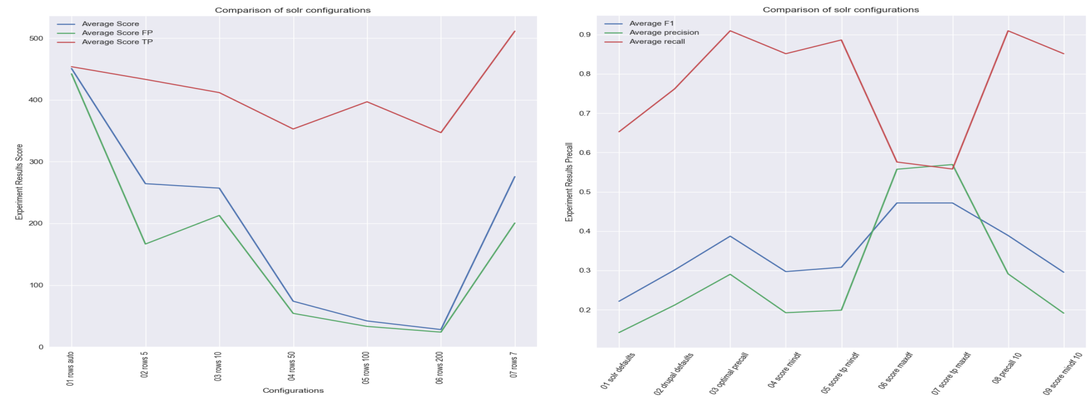
Method

A test dataset with

- Solr index made up of 1107 puzzle files
- 90 known duplicates that correlate with 32 duplicate groups

was used to

- Optimize the Solr index analyzer
- run automated experiments.
- Over 1000 search results and statistics, for example:
  - Average result placements and scroes
  - Average precision, recall and F1



Examples of automated Solr config experiments testing settings for MLT. Left shows how the average Solr Scores change with different number of row. The right shows average recall, precision and F1 while searching for the optimal configuration.

Average Optimal Settings for Solr MLT

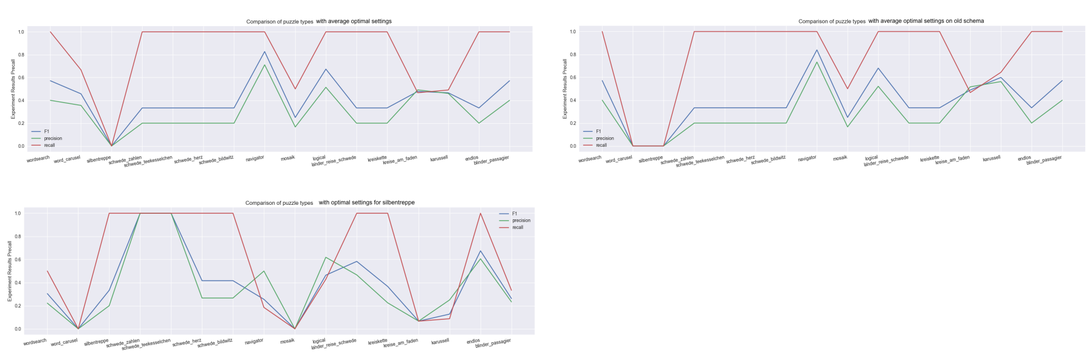
The focus was on recall. The experiments have shown that the following settings to deliver the best results with an average F1 of 0.38, average precision of 0.29 and average recall of 0.91:
"rows": 5, "mintf": 1, "mindf": 6, "maxdf": 10, "minwl": 5, "maxwl": 6, "maxqt": 200, "boost": false

Results

The developed results show that depending on the puzzle type, with a good Solr config and schema, good near duplicate search results can be achieved with MLT. Open thesis: maxtf setting or BM25 ranking function could improve the search results, but both are not available for MLT.

Due to the diversity of the documents and the high variance in the score of the search results no fully automated process can be reliably implemented.

The customer Rätsel Krüger is satisfied with the results. A list that shows five possible duplicates when new files are uploaded and he only needs to select the true duplicate groups helpes him a lot and will reduce manual workload.



Results evaluation by puzzle type. Top left shows average optimal settings, bottom left shows optimal settings for puzzle type silbentreppe. Top right shows average optimal settings before optimizing solr schema.

Project made possible with the help of
RÄTSEL-KRÜGER brightsolutions web.mobile.innovation

TU Graz – Institute of Interactive Systems and Data Science
8010 Graz, Inffeldgasse 16c, Austria, Tel.: +43 (316) 873-5624
office.isds@tugraz.at, www.isds.tugraz.at