# Recognition and Enrichment of Archival Documents

# D7.14
# Keyword Spotting Engines: QbE, QbS P1

Ioannis Pratikakis, Konstantinos Zagoris DUTH
George Retsinas, George Sfikas, Basilis Gatos, George Louloudis, Nikolaos Stamatopoulos NCSR
Alejandro H. Toselli, Joan Puigcerver, Enrique Vidal, Verónica Romero, Joan A. Sánchez, UPVLC
Tobias Strauß, Gundram Leifert, Roger Labahn, URO

Distribution:

http://read.transkribus.eu/

---

**READ**
**H2020 Project 674943**

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 674943

| | |
|---|---|
| **Project ref no.** | H2020 674943 |
| **Project acronym** | **READ** |
| **Project full title** | **Recognition and Enrichment of Archival Documents** |
| **Instrument** | H2020-EINFRA-2015-1 |
| **Thematic Priority** | EINFRA-9-2015 - e-Infrastructures for virtual research environments (VRE) |
| **Start date / duration** | 01 January 2016 / 42 Months |
| | |
| **Distribution** | Public |
| **Contractual date of delivery** | 31.12.2017 |
| **Actual date of delivery** | |
| **Date of last update** | |
| **Deliverable number** | D7.14 |
| **Deliverable title** | Keyword Spotting Engines: QbE, QbS P1 |
| **Type** | Demonstrator |
| **Status & version** | |
| **Contributing WP(s)** | WP7 |
| **Responsible beneficiary** | DUTH |
| **Other contributors** | NCSR, UPVLC, URO |
| **Internal reviewers** | Joan Andreu Sánchez |
| **Author(s)** | Ioannis Pratikakis, Konstantinos Zagoris DUTH<br>George Retsinas, George Sfikas, Basilis Gatos, NCSR<br>Alejandro H. Toselli, Joan Puigcerver, Enrique Vidal, Verónica Romero, Joan A. Sánchez, PVLC<br>Tobias Strauß, Gundram Leifert, Roger Labahn, URO |
| **EC project officer** | |
| **Keywords** | Keyword Spotting, Query by Example, Query by String |

# Table of Contents

## Executive Summary

Handwritten keyword spotting is the task of detecting query words in handwritten document image collections without involving a traditional OCR step. Recently, handwritten word spotting has attracted the attention of the research community in the field of document image analysis and recognition since it has been proved to be a feasible solution for indexing and retrieval of handwritten documents in the case where OCR-based methods fail to deliver proper results. This deliverable reports on the achievements concerning the tasks of keyword spotting for handwritten document image collections at the end of the second year of the READ project that have been realized by four (4) distinct frameworks which correspond to partners DUTH, NCSR, UPVLC and URO, respectively.

# I. The Query by Example (QbE) Engines

## 1. Introduction

A promising strategy to deal with unindexed documents is a keyword matching procedure that relies upon a low-level pattern matching called word spotting by example [Manmatha1996]. In the literature, word spotting appears under two distinct strategies wherein the fundamental difference concerns the search space which could be either a set of segmented word images (segmentation-based approach) or the complete document image (segmentation-free approach). The selection of the segmentation-based strategy is preferred when the layout is simple enough to correctly segment the words while the segmentation-free strategy performs better when there is considerable degradation on the document which is the common case in historical documents. Nevertheless both strategies use an operational pipeline where feature extraction and matching have prominent roles.

## 2. DUTH Keyword Spotting Framework

During the second year of the project, DUTH focused on two aspects in parallel : (a) Segmentation–free keyword Spotting in a Query by example framework and (b) a keyword spotting demonstrator that uses the aforementioned method to showcase its potential.

### 2.1. Segmentation-Free Keyword Spotting

The focus of the work during the second year has been to minimize memory and computational power requirements. That was of high priority since it would enable us to search in large document collections. The current method provides some unique advantages that stems from the capacity to search the whole document and not just applying a word segmentation method. Those advantages are:

1. Good handling of complex document layouts
2. Ability to match partial words or phrases
3. It can locate not only words but also symbols.

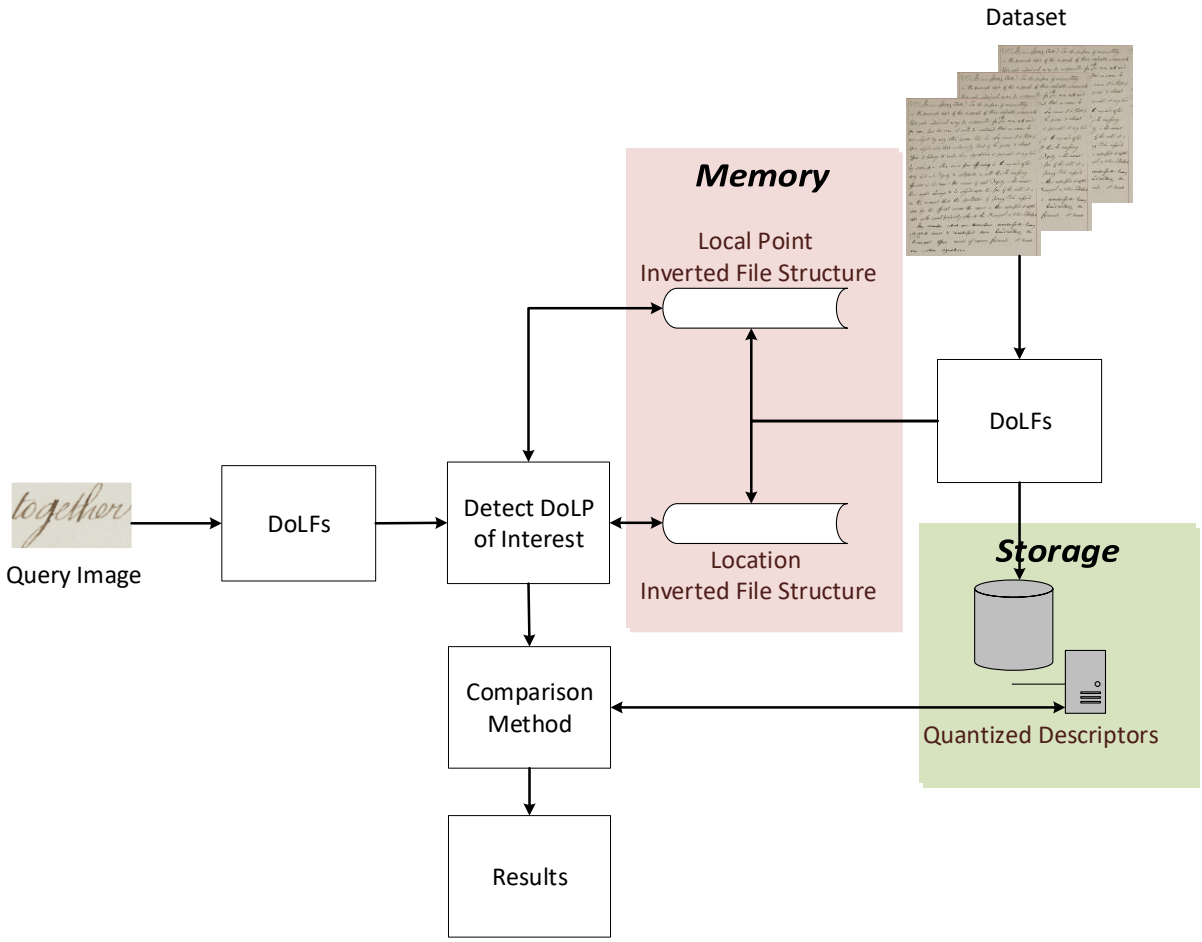



**Figure I.2.1** The architecture of the DUTH Segmentation-Free Keyword Spotting

Figure I.2.1 shows the architecture of the method. It uses Document Oriented Local Points (DoLFs)[ZAG2017] to detect meaningful points on a dataset and two types of Inverted File Structures to describe them. These two inverted file structures are the only required memory-based data since the DoLF descriptors are quantized to 64 bytes and stored in a storage database.

When the user searches for a word, the DoLFs are calculated and based on these two structures the most meaningful DoLFs are identified and retrieved from the storage. Finally, for comparative purposes the efficiency of the proposed method compared to the original method [ZAG2017] is used to draws the final conclusions. Section 4.2 describes the experimental results.

## 2.2. KeyWord Spotting Demonstrator

In order to showcase the above segmentation-free word spotting method, a cross-platform word-spotting application was created. It is based on Angular 5, Material Design and Electron frameworks for the front-end (GUI) and the back-end is created by the C#/.NET Core framework.

The KeyWord Spotting Demonstrator supports the following main tasks:

- Creation (Indexing) of new Datasets
- User interactive word image query selection
- Presentation of the spotted words

Figure I.2.2 shows some representative screenshots.

Moreover, the communication between the front-end and the back-end is defined by a REST API which is freely available at:

https://github.com/Transkribus/WSBackend-API

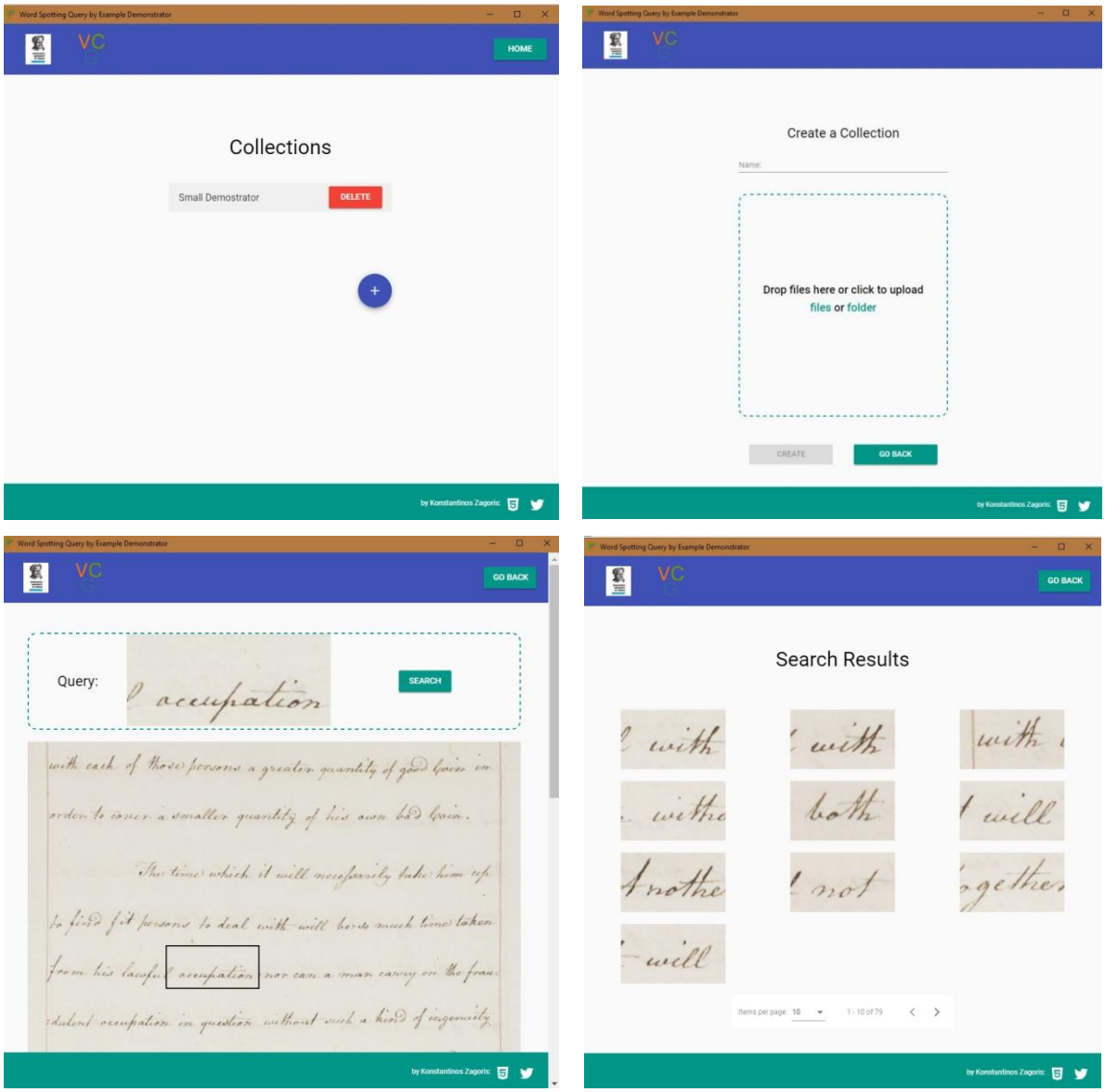



**Figure I.2.2** Screenshots of the KeyWord Spotting Demonstrator

# 3. NCSR Keyword Spotting Framework

On the second year of the READ project, NCSR focused on further enhancing the performance of QbE segmentation-based scenario. We also evaluated our methods on the segmentation-free scenario, which better simulates an end-to-end QbE scenario. Currently, our focus has shifted to Deep Learning and Convolutional Neural Networks due to their outstanding performance in various tasks.

## 3.1. Segmentation-based Word Spotting

At the end of the first year of the READ project, the best performance for the segmentation-based QbE word spotting scenario was achieved by the *NCSR-SeqPOG (M24)* method (see D7.13). This method consisted of the following steps: 1) main-zone detection [RET2016] 2) zoned feature extraction as sequence of features 3) a novel sequence matching based on dynamic programming. We have noticed that a critical performance factor was the first step, since an erroneous main-zone detection can significantly affect the overall performance of the system. To this end, we introduced different possible main-zones (multi-hypothesis) and we incorporated this extra information on the matching step, as an efficient variation of the previous matching algorithm. The new method, referred as *NCSR-MSeqPOG (M24)*, provides a notable boost in performance [RET2017a] (see section 4.1).

Time and memory requirements for the segmentation-based scenario are presented in detail at the deliverable of the first year (see D7.13).

## 3.2. Segmentation-free Variation

Segmentation-based scenario is very helpful at developing reliable word image representations, but does not correspond to a practical word spotting application. However, generating candidate regions for word images is a fairly simple task and has proven a compelling alternative to time-consuming segmentation-free techniques. Therefore, in order to apply the NCSR KWS methods on the segmentation-free scenario, a segmentation pipeline is used. For simplicity, we used the NCSR segmentation pipeline (up to word level) from the first year of READ project (D6.10).

It should be noted that the aforementioned pipeline produces unique candidate regions, which ideally correspond to the actual words of the document. However, this hard assignment of document image parts to words is subject to possible errors which affect the final word spotting performance. The aforementioned erroneous procedure can be avoided by using multiple hypotheses of candidate regions for each word starting from the initial document image. This idea is presented in D6.11 and will be explored during year 3 of the READ project.

## 3.3. CNN-based Word Spotting and Future Direction

Due to the immense research progress on Deep Learning and Convolutional Neural Networks, NCSR has already began to develop such techniques on keyword spotting and our plan is to replace handcrafted features with features extracted from CNNs. One of our main concerns is to significantly reduce the feature-vector size (dimensionality reduction) of such CNN features. This is a key step for enabling large scale applications and efficient indexing techniques. Towards this end, we have some preliminary results ([RET2017b], [RET2017c]), but we aim to further explore this direction during the third year of the project.

### 3.4. Implementation Observations

So far NCSR focused on developing well-performing methods that are also efficient in terms of complexity. Even though NCSR methods efficiency in terms of time and memory is noteworthy, there is still room for improvement (e.g. use of indexing for increasing the retrieval speed and dimensionality reduction for decreasing the storage requirements). This will be addressed at the third year of READ project.

## 4. Evaluation

The presented methodologies are evaluated against three datasets (Figure I.4.1):

- English Dataset which contains 115 Pages and 15923 words
- Konzilsprotokolle (German) Dataset which contains 100 Pages and 15579 words
- Finnish Dataset which contains 56 pages and 16201 words

Please note that the punctuation marks and capitals are considered in the ground truth corpora.

The queries consist of every word with length greater than 3 and frequency greater than 2. Therefore, the English dataset query set size is 4790, Konzilsprotokolle dataset query set size is 7119 and the Finnish is 5731.

The performance of the word spotting methods was recorded in terms of the Precision at Top 5 Retrieved words (P@5) as well as the Mean Average Precision (MAP) [Pratikakis2014]. Time and memory requirements are recorded in terms of the following metrics: Retrieval Time per Query, Memory requirements per Document, and Storage requirements per Document.

The evaluation of DUTH and NCSR methods is performed on an 8-core Intel i7-4770K at 3.50GHz with 16Gb of RAM for parallel computation (4 cores). All DUTH methods are currently implemented in C#/.NET. All NCSR methods are currently implemented using MATLAB.

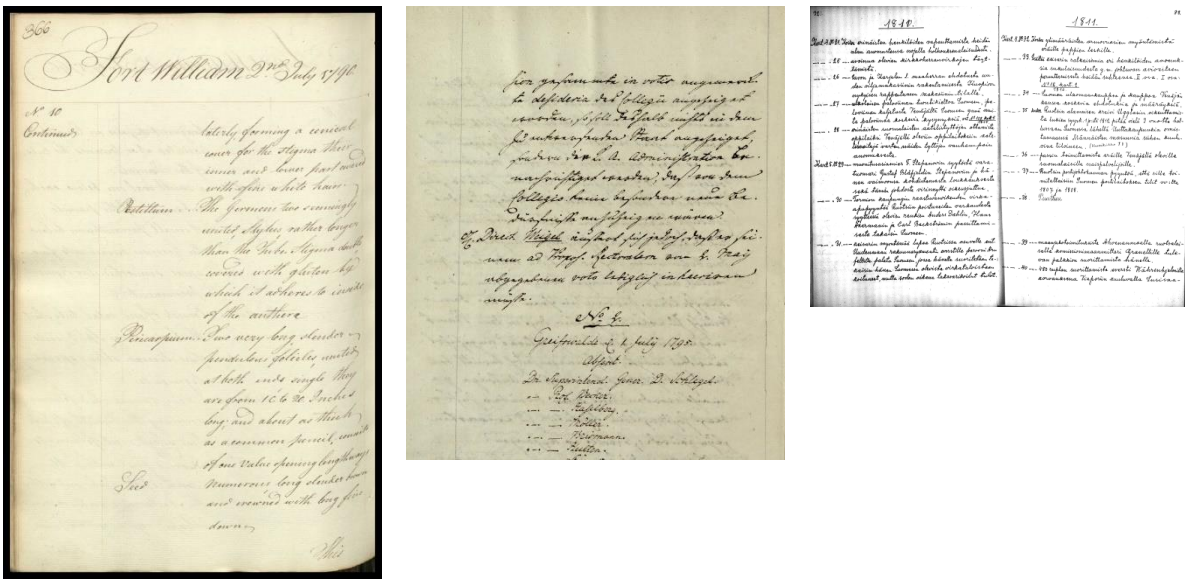


**Figure I.4.1** Example documents from the English (left), Konzilsprotokolle (middle) and Finnish (right) Datasets

## 4.1. Conclusive remarks on the Segmentation-Based Scenario

Table I.4.1 contains the performance results on the segmentation-based scenario for DUTH and NCSR methods. We observe that the new NSCR method (NCSR-MSeqPOG (M24)) has a constant gain in performance compared to previous methods. Nevertheless, it should be mentioned that as stated in Section I.2.1 the focus of DUTH during the second year has been to minimize memory and computational power requirements as a trade-off with effectiveness.

**Table I.4.1** Experimental Results Segmentation-Based Evaluation

| Method | English | | Konzilsprotokolle | | Finnish | |
|---|---|---|---|---|---|---|
| | P@5 | MAP | P@5 | MAP | P@5 | MAP |
| **Original[ZAG2017]** | 0.56 | 0.42 | 0.78 | 0.64 | - | - |
| **DUTH M12** | 0.42 | 0.33 | 0.54 | 0.38 | - | - |
| **NCSR-ZAH** | 0.55 | 0.41 | 0.69 | 0.52 | 0.63 | 0.50 |
| **NCSR-POG (M12)** | 0.55 | 0.42 | 0.73 | 0.58 | 0.71 | 0.65 |
| **NCSR-SeqPOG (M24)** | 0.63 | 0.49 | 0.81 | 0.68 | 0.80 | 0.77 |
| **NCSR-MSeqPOG (M24)** | 0.64 | 0.51 | 0.82 | 0.71 | 0.81 | 0.79 |

## 4.2. Conclusive remarks on the Segmentation-Free Scenario

Table I.4.2.1 shows the segmentation-free evaluation results for the original method [Zagoris2017], as well as methods 'DUTH-M12', the new 'DUTH-M24' and 'NCSR-POG (M12)'. The time, memory and storage requirements are presented in Table I.4.2.2 by averaging the corresponding metrics over the three datasets.

**Table I.4.2.1** Experimental Results Segmentation-Free Evaluation

| Method | English | | Konzilsprotokolle | | Finnish | |
|---|---|---|---|---|---|---|
| | P@5 | MAP | P@5 | MAP | P@5 | MAP |
| **Original[ZAG2017]** | 0.35 | 0.22 | 0.59 | 0.42 | 0.58 | 0.43 |
| **DUTH M12** | 0.38 | 0.25 | 0.46 | 0.24 | 0.35 | 0.23 |
| **DUTH M24** | 0.34 | 0.22 | 0.51 | 0.27 | 0.55 | 0.35 |
| **NCSR-POG (M12)** | 0.36 | 0.36 | 0.64 | 0.54 | 0.67 | 0.62 |
| **NCSR-SeqPOG (M24)** | 0.44 | 0.42 | 0.77 | 0.66 | 0.76 | 0.75 |

Table I.4.2.2 shows that 'DUTH-M24' method manages to keep the same or in some cases better performance than 'DUTH-M12' with a big reduction in memory requirements enabling the capability to search in large datasets. Moreover, 'DUTH-M24' has the same performance with our original method at the P@5 in much less query time and memory requirements.

Table I.4.2.3 shows the performance of 'DUTH-M24' in relation to the dataset size. The results reveal that the retrieval time per query is increased in a non-linear manner so that make search feasible in terms of time consumption for large scale datasets.

Table I.4.2.1 shows that both NCSR methods manage to achieve outstanding performance, while NCSR-SeqPOG (M24) outperforms any other method. It should be mentioned that there is still room for improvement since the default segmentation pipeline of the first year is used (see section 3.2). At the same time, NCSR methods have very low time and storage requirements (see Table I.4.2.2), which indicates that they are suitable for large scale applications. Nevertheless, NCSR methods have not been optimized implementation-wise. Their requirements can be further reduced using indexing and dimensionality reduction techniques, respectively.

**Table I.4.2.2.** Time, Memory and Storage Requirements for Segmentation-free Scenario

| Method | Retrieval Time per Query (sec) | Memory requirement per Document (KB) | Storage requirement per Document (KB) |
|---|---|---|---|
| **Original** | 15.84 | 19800 | 19800 |
| **DUTH M12** | 0.36 | 1410 | 1410 |
| **DUTH M24** | 0.67 | 366 | 2187 |
| **NCSR-POG (M12)** | 0.0080 | 101 | 101 |
| **NCSR-SeqPOG (M24)** | 0.0653 | 424 | 424 |

**Table I.4.2.3.** Comparative Evaluation Results for big datasets for Segmentation-free Scenario using **DUTH-M24** method

| Dataset (Documents) | Retrieval Time per Query (sec) | Overall Memory requirement (MB) | Overall Storage requirement (MB) |
|---|---|---|---|
| **50** | 0.47 | 12 | 35 |
| **5000** | 1.55 | 1125 | 3438 |
| **50000** | 3.21 | 11648 | 34966 |

# II. The Query by String (QbS) case

For a set of text images, keyword spotting (KWS) consists in finding the images (and maybe the regions or locations within each image) where specific words may appear. Rather than deterministic results, KWS systems are expected to provide, for each detected spot of a query word, a confidence score which measures how sure is the system that the word appears in the spotted image or location. This allows the user to somehow establish a confidence threshold to specify the required "precision-recall trade-off"; that is the balance between the accuracy of the spotting results (referred to as "precision") and the number of correct images retrieved (referred to as "recall").

In the QbS KWS setting, query words are given in the form of strings of letters, which is a very flexible and convenient form in many applications. Also for this very same reason, QbS KWS properly provides the basic technologies to develop indexing and search systems which aim at supporting fast free-text content access to (very) large collections of untranscribed handwritten text images.

# 1. UPVLC Keyword Spotting framework

UPVLC develops QbS KWS technologies within the information-retrieval domain and following well-funded statistical methodologies. The spotting confidence score is assumed to be the probability that an image, region, or location is "relevant" for the query keyword. An image region or location is considered to be relevant if the word(s) written in it honor the query. Following this very general framework, several approaches are being developed by UPVLC. These different approaches aim at properly dealing with corresponding indexing and/or search problems raised by indexing and search applications involving hundreds of thousands or even millions of handwritten page images.

The work carried out by UPVLC under this framework during the second year of READ is described in the following subsections. Each of the two first subsections is associated with a publication in a scientific journal or in the proceedings of a major international conference. Therefore, only a brief summary of each work is provided, accompanied by the corresponding reference to the published paper.

In addition, in Section 1.3 we describe work carried out towards a standardization of architecture, tools, formats and work flow in the use of QbS KWS for probabilistic indexing purposes.

## 1.1. Simple and Effective Multi-Word Query Spotting in Handwritten Text Images

Keyword spotting techniques are used to develop cost-effective solutions for information retrieval in handwritten documents. We explore the extension of the single-word, line-level probabilistic indexing approach described in our previous works to allow page-level Boolean combinations of several single-keyword queries. We propose heuristic rules to combine the single-word relevance probabilities into probabilistically consistent confidence scores of the multi-word boolean combinations. As a preliminary study, this work focused on evaluating the search performance of word-pair queries involving just one OR or AND Boolean operation. Empirical results of this study support the proposed approach and clearly show its effectiveness.

See details in [NOY2017].

This work has recently been extended with new empirical evaluation results which complete some of the points which were left open in [NOY2017]. This extended work has been submitted for publication in the journal "Pattern Analysis and Applications".

## 1.2. KWS Experiments for Large-Scale Indexing of a Vast Medieval Manuscript Collection in the HIMANIS Project

A vast medieval manuscript collection, written in both Latin and French, called "Chancery", was considered for indexing at large. In addition to its bilingual nature, one of the major difficulties of this collection is the very high rate of abbreviated words which, on the other hand, are completely expanded in the ground truth transcripts available. Before undertaking full indexing of Chancery, experiments were carried out on a relatively small but fully representative subset of this collection. To this end, a keyword spotting approach has been adopted which computes word relevance probabilities using character lattices produced by a recurrent neural network and a N-gram character language model. Results confirmed the viability of the chosen approach for the aimed large-scale indexing and showed the ability of the proposed modeling and training approaches to properly deal with the abbreviation difficulties mentioned.

See details in [BLU2017].

Recently, the real indexing of the complete Chancery collection has been very successfully accomplished. It encompasses about 200 bundles with 83,000 page images. The process required about 1 month of intensive multi-core computation and the resulting probabilistic index contains about 300 million entries and requires about 12 gigabytes of storage. During this process, more than three million lattices were generated, then used to compute the probabilistic index entries, and finally discarded. All in all, this workflow involved handling about 300 gigabytes of data during the whole process time span of about two months. A beta version of the query and search system for the full Chancery collection is available at http://prhlt-kws.prhlt.upv.es/himanis.

This work was mainly funded by another project (HIMANIS) but it was also significantly supported by READ. In particular, this work constituted a first large-scale test-bed for the probabilistic indexing standardization guidelines, developed in READ and discussed below (Section 1.3).

## 1.3. Proposal to standarize Architecture, tools, workflow and index formats for probabilistic keyword indexing and search.

After a series of discussions in READ, the following two documents were produced:

- "Proposal to standardize keyword indexing and search in READ and Transkribus" [VID2017].
- "Proposal to standardize keyword indexing and search in READ and Transkribus. Workflow of the Indexing Tool" [TOS2017].

Examples of collections indexed by PRHLT-UPVLC following the guidelines of this proposal:

- PLANTAS Vol VII (about 700 images, from the tranScriptorium project, 2013-2015). http://transcriptorium.eu/demots/kws/index.php/ui/chapters/plantas
- Passau small miscellaneous collection (READ -- Work in progress; about 90 images as of oct-2017). http://transcriptorium.eu/demots/kws-Passau

- Siglo de Oro -- Spanish Theater Golden Age (READ -- Work in progress; about 2,700 images as of nov-2017).
  http://transcriptorium.eu/demots/kws-Lope
- Chancery (about 83,000 images from the HIMANIS project, 2016-2017).
  http://prhlt-kws.prhlt.upv.es/himanis
- Other collections (2010-2017).
- http://transcriptorium.eu/demots/KWSdemos

# 2. Rostock Keyword Spotting framework

Keyword spotting as it is implemented in Transkribus is strongly related to the hadwritten text recognition (HTR). The output of the neural network (the so-called *ConfMat*) is saved and used for keyword search. In fact, keyword spotting is more reliable than transcription: The results of the search are manually verified, false positives are just ignored. In contrast to the transcription, uncertainties of the recognition process are taken into account.

## 2.1. Workflow

The first steps of the workflow equal those of the HTR process (compare deliverable D7.8):

**Text line extraction:** As a very first step, the page image is processed by the line segmentation tool (see deliverable D6.11) which provides separated text lines.

**Indexing:** These text line images are processed by the HTR. The HTR is trained to output a probability for any character per position (see deliverable D7.8). For a whole text line, these probabilities form the ConfMat. These ConfMats are saved for any text line image.

**Search / decoding:** Searching means to calculate the probability for the query word to appear in a ConfMat. If the probability exceeds a certain threshold, we say the query matches the corresponding text line.

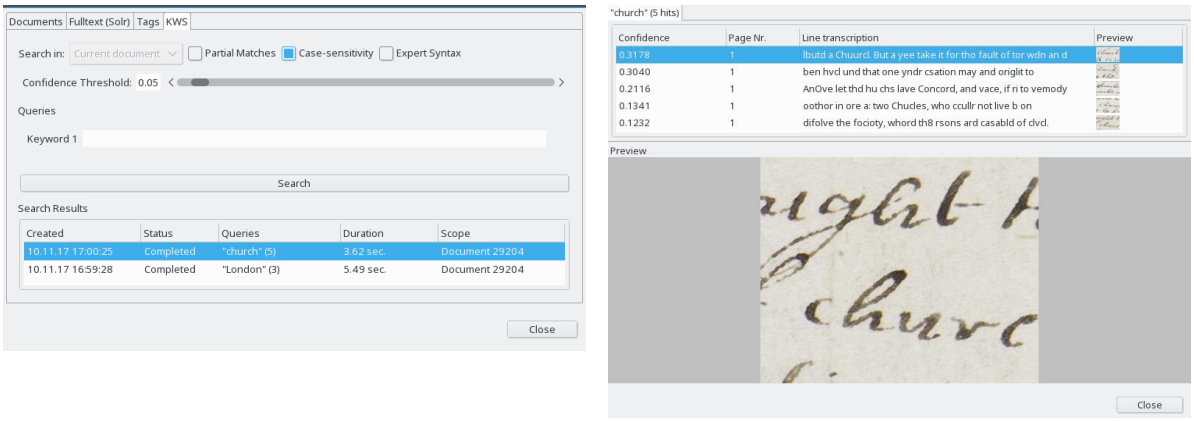
## 2.2. Setups

The classical KWS searches for an isolated occurrence of the query word within a text line. Besides this standard setup, there are modified setups for the keyword search (compare Figure II.2.1):

**Partial matches:** Search for any occurrence of the query word -- also as subword. For example, the query “key” would also match at “keyword”.

**Case insensitive:** Search for occurrences of the query word while considering spellings independly of upper-case or lower-case letters. For example, the query ``key'' would also match at “Key” or “KEY”'.

**Regular expressions:** Search for a more general pattern rather then single words, e.g., for dates of a certain time period or different spellings of the same word. These patterns have to be formulated as regular expressions. For example, the query [0-9]{4} representing four arbitrary digits would match any year.



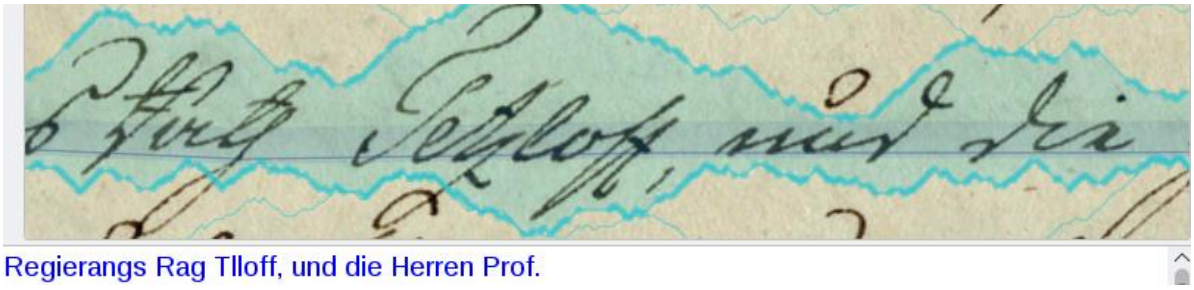

a) KWS menu with setups, query text box and previous searches

b) Results window of an exemplary search for the keyword ``church'' with several matches and a preview

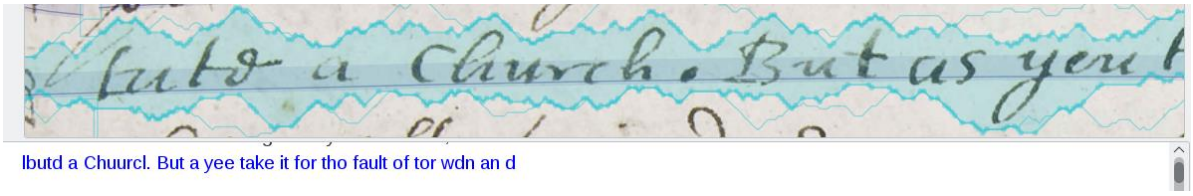
**Figure II.2.1** Operating elements of the KWS Implementation of Transkribus

## 2.3. Implementation

The keyword search is already implemented and available to authorized users. Figure II.2.2 shows the operating elements of the KWS implementation of TranskribusX.




a) Query: Tetzloff, transcription: Tlloff




b) Query: church, transcription: Chuurcl

**Figure II.2.2** Two examples for correct KWS matches with a bad transcription

# References


[BLU2017] T. Bluche, S. Hamel, C. Kermorvant, J. Puigcerver, D. Stutzmann, A.H. Toselli, E. Vidal, "Preparatory KWS Experiments for Large-Scale Indexing of a Vast Medieval Manuscript Collection in the HIMANIS Project", ICDAR, 312-317, 2017.

[NOY2017] E. Noya-García, A.H. Toselli, E. Vidal, "Simple and Effective Multi-word Query Spotting in Handwritten Text Images", Iberian Conference on Pattern Recognition and Image Analysis (IbPRIA), 76-84, 2017.

[RET2016] G. Retsinas, G. Louloudis, N. Stamatopoulos and B. Gatos, “Keyword Spotting in Handwritten Documents using Projections of Oriented Gradients”, Workshop on Document Analysis Systems, pp. 411-416, Greece, 2016.

[RET2017a] G. Retsinas, G. Louloudis, N. Stamatopoulos and B. Gatos, “Efficient Learning-Free Keyword Spotting”, IEEE Transactions on Pattern Analysis and Machine Intelligence, Under Review.

[RET2017b] G.Retsinas, G.Sfikas and B.Gatos, "Transferable Deep Features for Keyword Spotting", International Workshop on Computational Intelligence for Multimedia Understanding, EUSIPCO 2017

[RET2017c] G.Retsinas, G.Louloudis, N.Stamatopoulos, G.Sfikas and B.Gatos, "Nonlinear Manifold Embedding on Keyword Spotting using t-SNE", International Conference on Document Analysis and Recognition (ICDAR), 2017

[TOS2017] A.H. Toselli and E. Vidal "Proposal to standarize keyword indexing and search in READ and Transkribus. Workflow of the Indexing Tool". Technical report, September 2017. READ Wiki: https://read02.uibk.ac.at/wiki/index.php/KWindexing

[VID2017] E. Vidal Proposal to standarize keyword indexing and search in READ and Transkribus. Technical report, February 2017. READ Wiki: https://read02.uibk.ac.at/wiki/index.php/KWindexing

[ZAG2017] Zagoris K, Pratikakis I, Gatos B. Unsupervised Word Spotting in Historical Handwritten Document Images using Document-oriented Local Features. IEEE Transactions on Image Processing. 2017