Good afternoon everyone im very glad that you are able to join us today - even though the circumstances for hosting a workshop might not be ideal -- still very happy that we were able to set up this zoom environmnet

the pandemic really prepared us for unforseen circomstances like these - so we can quickly pivot to oonline.

Alright, my name is WOuter Haverals, i'm happy to be presenting today together my colleaguge Christine Roughan a workshop on Handwritten Text recognition. Christine and I are both postdocs working at the center for digital humanites here at princeton -- and our work involves turning handwritten material likek you see on the slide over here into typeset, machine readable text.

Our workshop today consists of two main parts. First i will talk a little bit about what HTR is, where it comes from and why we do it. And we'll very briefly look at two popular platforms for performing handwritten text recognition.

In the second component of this workshop, Christine will talk us through the actual process and workflow, gearing up for the hands-on exercise that we will carry out at the end.

Alright, let me start by asking you the simple question if you can read this? If you are an expert in 14th century medieval dutch paleography -- then this should be a walk in the park. If not: I can imagine that this is very hard to read for you.

The amazing thing is that it is not hard to read for Transkribus, whcic is one of the maim platforms to perform handwriotten text recognition -- and also the one we will be using later on in this workshop.

Handwritten Text Recognotion comes down to the process of turning handwritten material -- in different languages, different centuries -- into machine readable, typeset text, whcih can then be used for linguistic analysis, or it can serve as the basis for a printed edition of the text. or the text could accompany the digital surrogate that is served to you in an online viewer -- the possibilities are nearly endless.

---

however, already from the start i want to note that automatic Handwritten Text Recognition is not perfect. As you can see in this example here, there are some characters misrecognized, 4 out of the total 246 characters are either missed or incorrectly recognized. if we break it down into a ratio of the correct characters versus incorrect characters, you can say that for this example there is a Character Error Rate of 98 percent. Meaning that on average here 2 out of 100 characters are still recongnized incorrectly by the automatic htr method that was used here.

you can also look at this from the word level, saying that 4 out of the 55 words have some kind of mistake in them. So this means that the Word Error Rate is ca. 93 percent. This is a pretty good result, especially given that the handwriting is not thqt eqsy to read.

Alright, now, If we think about how much written text there is out there that is already available to us in a fmachine readable format -- i.e. computers can read it, and you can intereact with the text in a way as you would in e.g. Microsoft word, this is what we have: the tip of the iceberg. The biggest part is still very much invisible, it lives underneath the surface of the water. There is a simple reason for that: the majority of

handwritten sources are simply not digitized yet -- despite major efforts that people have been uinderaking in this area the past decades.

Since my background is in medieval studies, I like to refer to the example of the Vatican Library in this respect -- or as medeivalist call it: the Vat. The Vat is among the biggest mansucript libraies in the world - especially if we also count their archive. The manuscript collection alone holds ca. 80,000 books or codices from both the medieval and the humanistic period. A very rough estimate provided by the Vat itself says that this amounts to approximately 40 million pages.

The Vatican Library started digitizing its collection in 2010! So about 14 years ago. By digitization, in this case i mean they started taking digital photographs of all of the manuscript pages in their collection -- and subsequently serving them over the internet to the community for everyone to see and explore.

In terms of storage, this is also quite the Herculean task: they estimate that they will need about 45 million gigabytes of harddisk storage. Of course this kind of storage is really nothing compared to the actual physical storage capacity they need to store all these manuscrips -- we're talking miles and miles of shelves here.

alright, so they started with this endeavor some 10 years ago. Does anybody want to guess where they are with this task right now? How many of the 80,000 manuscripts have been fully dogitized? I an already tell you that they arent done yet.

Yeah, so they are not even at the halfway point yet. They have roughly digitized 27 thousand manuscript! They are not lazy or slow workers, it is an incredibly laboriius task + technology changes. The way digitization was done 10 years ago is not the same today!!

One year ago – when I gave a similar version of this presentation: 22,910 manuscripts digitized.

But... only pictures. NO TEXT!

This is because by themselves, computers cannot "read" the same way humans can. Computers have to break down an image into segments, which are then turned into strings of zeros and ones -- which computers CAN read computers and interpret.

In very broad strokes, this is what happens in the background of a computer program that is developed for the purpose to read the text on an image.

Now, there are two different flavors so to speak of these computer programs or technologies to read text; they are very different in how they process and image and consequently what you can do with it.

The first one is called Optical Character Recognition or OCR. Traditional OCR breaks words down an image into a series of smaller letter-images by looking for the spaces between letters. It then compares each letter-image to a database of letter layouts. Then the program decides which letter best matches the image. Finally the software translates the image of the letter into a machine readable, thereby making the text searchable.

Nowadays, OCR is considered more or less "solved" in computer science. There are still improvements to be made, and there are still languages for which OCR is underdeveloped -- but overall OCR is solved, reaching character accuracies of more than 99 percent. This is mainly becasue letter forms are standardized, and the backgrounds are generally very clear.

OCR software, however, really only works on typeset text. It's lousy for anything written by hand.

The main problem in this example is the lack of space between letters (so-called dirty segmentation). OCR can't tell where one letter stops and another starts, and therefore doesn't know how many letters there are. It's a catch-22 that has a specific name in computer science: Sayre's paradox.

> The result is a computational deadlock, sometimes referred to as Sayre's paradox: OCR software needs to segment a word into individual letters before it can recognize them, but in handwritten texts with connected letters, the software needs to recognize the letters in order to segment them. It's a catch-22.

There are other issues as well: mostl importantly, since everyones writing style is unique, there is a virtually infinite number of writing styles. Another issue is the background of the image: many handwritten documents have very noisy backgrounds, think of medieval manuscript which is organic material and for that reason show a lot of imperfections.

Finally, the human writing process itself can also be noisy, we tend to scratch out words, disrupt the linearity of the text by putting notes in the margin, below or above the text where it should be inserted.

In short: OCR is not made to deal with these obstacles!

This is where the technology of Handwritten Text Recognition or HTR stept in and can make a big difference -- turning thousands of scanned handwritten pages into text.

HTR software works very differently from OCR -- instead of pattern matching, HTR relies heavily on machine learning or artificial intelligence. It is a program that needs to be trained on the basis of examlpes of the handwriting joined by the actual images of the handwriting. In a brief moment, Christine will tell you more about this process in detail.

At this point it is important to note that -- among some other initiatives -- there are two major platforms whcih both really excell at HTR, and that have a critical mass of users: Transkribus and Escriptorium.

Transkribus is originally developed by the University of Innsbruck as part of the READ project. A first verison was released in 2013, and ever since there have been a lot of updates and treansformations. Today, it is mostly a web-based platform that excels in recognizing Western European languages, particularly Latin scripts.

On the other hand, there is eScriptorium, whcih is developed by the Université Paris Science et Lettres (PSL) and first released in 2018. eScriptorium offers both web services and local installation options,

making it less dependent on internet connectivity, however it is harder to install on your local machine than Transkribus -- which is the main reason why in the remainder of our workshop we will look at Transkribus. EScriptorium is known for its versatility with diverse scripts, including non-Latin, such as hebrew and Arabic.

A stark difference between the two is that we don't really know what kind of algorithm is at the heart of Trnakribus. The source code is closed off to the public, the reason for this is their payed subscription. Transkribus requires you to make an account and select a monthly plan -- there is a free plan, which gives you access to limited options though.

eScriptorium, on the other hand is open source, you can adapt the code yourself, if this is needed for your project. It is also free, the only drawback is that you have to rely on the compute power of your own machine. Wherease with Transkribus, you are basically paying to make use of their infrastructurte for the process of machine learning.

Other than that there arent a lot of differences -- both platforms use advanced HTR technology to facilitate for the transcription of historical texts.

So, at a very general level: how do you do HTR? From a macro perspective the workflow looks like this, and Christine will go into this at a much more detailed level, taking into account different requirements:

First of all, you need a digital image and upload it to the platform of your choice.

Next, the input images have to be prepared. Generally, this includes converting it into greyscale, and making sure it looks nice overall, sometimes dewarping or denoising the image

Next, the layout has to be analysed: basically, this comes down to drawing boxes around the text regions, and spereating them from the non-text areas, saying to the computer: look here for the text.

Next comes the actual text recognition.

Finally, there is the optional step of correcting some of the mistakes inevitably still be part of your result.

Alright, I am going to hand it over to Christine now, who will