Inhalt / Content

Machine Learning sounds like a promising new world for literary studies. Sounds like we can put something into our computers, press a button and then only have to interpret consistent data… But when you start to really get into it and apply the procedures, you soon realise, that there is not much consistency in literary studies. The problem is that there is still far too little training material to teach our computers to read literature. So we have to create training data ourselves. But what must it look like? To answer this question I have made a few tests. Today, I would like to share the results of these texts with you.

What is training data

The term training data sounds dry, clinical and somehow not humanistic. Besides, it is more than just a little misleading. It sounds as if there is a lot of data somewhere that just needs to be picked up. Well, that's deceptive. Training data for machine learning is, simply put, a collection of examples of language phenomena. With these examples you can teach an algorithm to learn how to recognize a particular language based category of data.

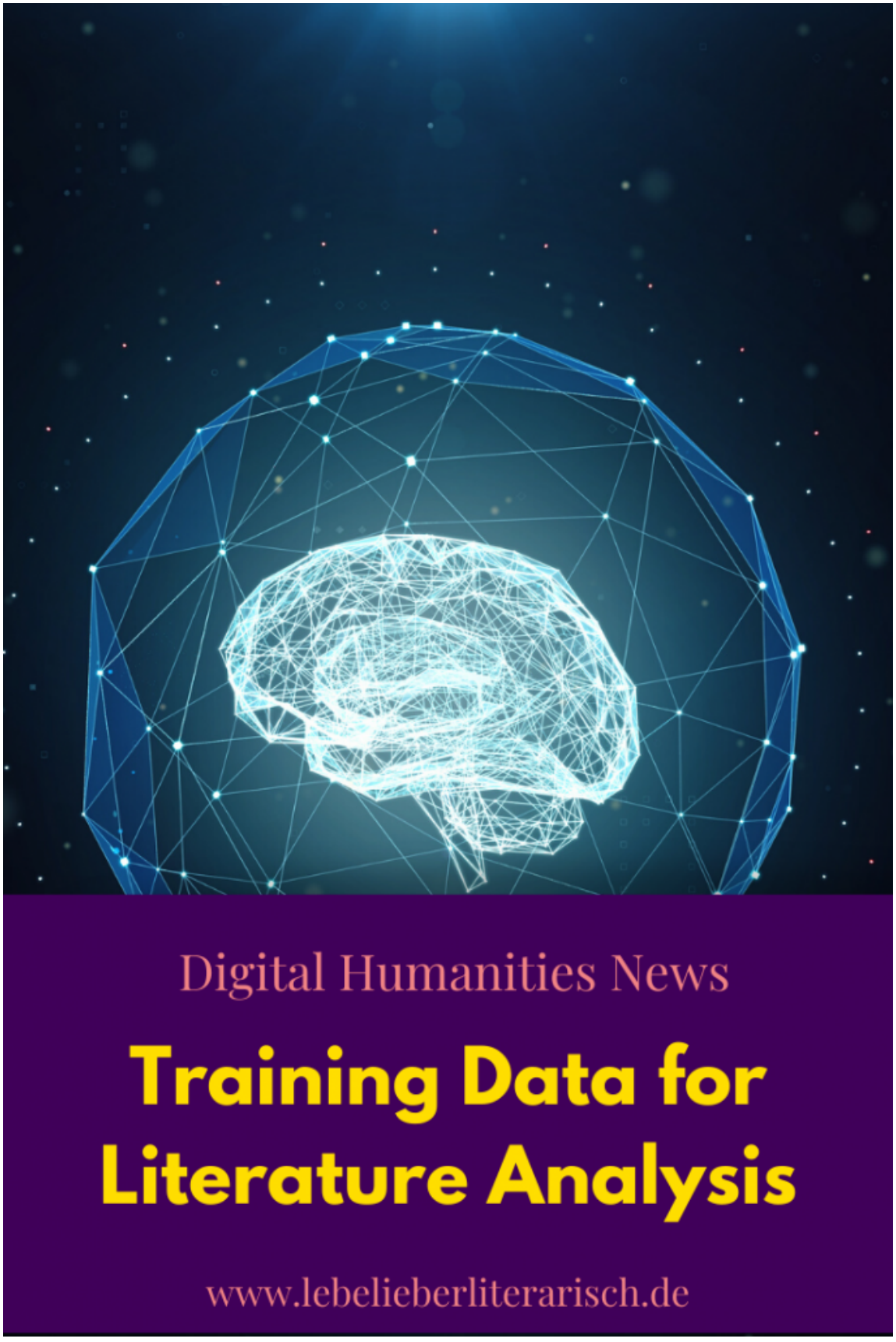
There is this word "data" again, which in principle still leaves unclear what I am talking about. So, one example: In my doctoral thesis I'm investigating the literary category of space. For this purpose I train a machine learning tool in such a way that it (hopefully) automatically recognizes spatial expressions. So my training data are literary texts in which

I have marked room expressions. The algorithm of the tool can compare these examples and learn from them. In the end it hopefully knows what I understand by room expressions in literary texts.

How can I create training data myself?

Well, if you read this blog often, you probably already know that I do not use pen and paper to collect room expressions in literary texts. Still, you might ask yourself how exactly my training data looks like. And this is where it gets a little tricky. Because the exact format of the data depends on the tool you want to use. The only common denominator is, that the data must be available in digital form.

This means, that if you want to create a training corpus for a machine learning tool yourself, you first have to find out what format the tool needs. Many tools I know need pure text formats like txt or csv. Some can also handle xml. Very common in the Digital Humanities is the TEI-xml format. This is a standardized variant of xml that allows a lot of metadata. It can thus store additional information in the text file without the tool counting it as primary text.



Annotate categories

If you have texts that are in the right format for your tool, the next step is to mark up categories the algorithm should learn. Let's stick to the example of my PhD thesis topic, the category of space. So I have to mark every room expression in the text as such. This is usually done in the form of a so-called inline markup. You insert short so-called xml-tags into the text, which show where a room expression starts and where it ends.

It is also possible that a tool can only evaluate training data in form of tables. This is the case, for example, with the Stanford Named Entity Recognizer (Finkel, Grenager and Manning, 2005). By the way I love to use this tool. I not only chose to work with it in my PhD, but also in the m*w project, which I have mentioned several times on this blog already.

Putting training data into the right form

For the StanfordNER, I first convert my training texts into one-word lists and then into a table. In this table words are put into the left column and in the right column there is an O behind each word. This stands for Other and must of course be replaced by the names of my categories, where spatial expressions appear. Or in the m*w project, in which we are investigating gender stereotypes, by the categories of female, male or gender-neutral, where character names appear (Schumacher, 2020).

What do I have to consider when compiling training data?

Depending on the goal you have in mind, your training data can look quite different. For example, if you want to save some time when analyzing a very extensive novel by having characters annotated automatically, it may be sufficient to prepare the first chapter of the novel as training data. Since the training text and the object of analysis are very, very similar (same writer, same novel, just different passages in the text) a machine learning tool can achieve quite good results with this.

But suppose you want to use Machine Learning to train a generic tool that you can apply to very different texts. Then it will be a little more difficult for the algorithm. A writer can use different spatial expressions in different texts. Other writers may use words of this this category even more differently. And in other centuries, there may have been not only different spellings for the same spatial expressions, but perhaps also completely different

terms to represent space. The solution for such complex use cases are large training corpora, which are cut together from several texts by different authors from different centuries.

Size of the training corpus

The size of your training corpus can actually be crucial. The more complex the categories you want to have automatically detected, the larger the training corpus should be. Fortunately, there are already some researchers working on domain adaptation of machine learning methods. That means they are testing what you have to do to be able to work with the tools originally developed in other disciplines, such as computational linguistics.

Jannidis et al. have found, for example, that a training corpus of 30,000-40,000 tokens is a good starting point for the automatic recognition of character names in narrative texts (Jannidis et al., 2015). From here, the training data can then be enlarged bit by bit (cf. ibid.). Tokens in this case refer to words in a continuous text, which means that words that can be mentioned more than once. When I train a machine learning tool, I therefore always start with a training corpus of this size.

Composition of training corpus

In their tests, Jannidis et al. put together training data from relatively short text sections. These were randomly selected from any part of the narrative texts used. I have also tried this procedure for my purposes, but found that it is less suitable for my categories.

In concrete terms, this means that for my location category, for example, I first put together a training corpus of 15 narrative texts. To get to 40,000 tokens, I copied a passage of 2666 words from each text. Sometimes from the beginning, sometimes from the middle, sometimes from the end – quite randomly. The results didn't knock me off my feet yet, so I tried something else. Since I suspected that spatial expressions are particularly common at the beginning of a narrative text, I tested what happens when I put together longer initial passages. I reduced the number of texts to 10 and took 4000 tokens from each text. Afterwards the result was much better for my categories.

How heterogeneous can my training corpus be?

Another question I asked myself when I started using machine learning methods for literary studies was, how much the texts in the training corpus may actually differ from the texts

you want to study. I knew that most off-the-shelf-tools are not optimized for applying them to literary texts. This is mostly due to the fact that they were mainly trained with data from factual texts (newspaper articles to be precise). I have already mentioned that these tools can work well when trained with excerpts from the examined text. But how well must the training corpus fit to the examined texts, if they differ fundamentally from the training corpus? This question is especially important if you want to train a generic tool.

Gender attributions – a category that allows heterogeneous training data

Well, the answer is once again: It depends on what you want to investigate. In the m*w project, in which my colleague and I train a named entity recognition model that can recognize male, female and neutral figure names, it has been shown that the training corpus can be wide. For our core corpus, which consists of novellas of the German "Novellenschatz", I once tested a training corpus that was compiled from other novellas of this collection, i.e. short narrative texts from the 19th century (altogether about 80,000 tokens). Then I created a training corpus of the same size, which was composed of longer narrative texts, mostly novels, from the 18th and 19th century. With both I achieved about the same results with 5 test texts (only one test text had a deviation of 4% in favor of the novella training corpus.

Space – a category that needs homogeneous training corpora

With my location categories it looked a little different. Here I created a training corpus consisting of longer narrative texts from the 18th century and one of equal numbers of texts of the same genre from the 19th century (both with about 40,000 tokens). For Schiller's "Geisterseher", the training data from the 18th century texts worked much better. The novella "the grey John" from the 19th century showed better results with the other training corpus. For the room category, which by the way has 5 subcategories and is therefore quite complex, training corpora that fit the object of investigation are better. That means, if you also try to use machine-learning methods for your research, it is definitely worth doing some tests to find out how heterogeneous your training data may be.

What is fuzzy training data?

In addition to the size and composition of training data, there is another pitfall in the use of machine learning methods in literary studies, the so-called "fuzzy" data. And although I

myself struggle with this problem in my projects, I would like to cite the project of a student from my last seminar as an example. It would be best if you read her own, very entertaining and informative essay (Benz, 2020). This example shows very well that we have to rethink something when we work with machines.

This is because, unlike us humans, complex categories such as “court system” (the category Nele wanted to work on), which can include figures as well as buildings, documents and descriptions, are too intangible for computers. They are too fibrous, just “fuzzy”. When dealing with such a supercategory, it is worth thinking about precise subcategories, such as “legal person”, “legal document”, “legal building”, etc. But, I already hinted above, the more subcategories you have, the more likely it is that your training corpus will have to be large. After all, the computer needs enough examples for each category to learn.

Does this really get me to the point where I no longer have to read myself?

You’re probably asking yourself, “Is it all worth it?” Maybe you think you might as well wait until someone else has optimized the tools you want to use for literary studies. And indeed, the road to reliable automatic recognition of literary-scientific categories in texts is still a long one. For getting to the point of being sure that you can apply your machine learning tool to a text without missing crucial information, you will have to invest many hours, weeks or even months.

But what is crucial for me is that this time is not lost. Because in this process the computer is not the only one learning something. You will get to know your categories better. You will recognize where you need to sharpen your assumptions. And you will learn about your training data and test texts. If you know how to use this part of the cognition process for yourself, I can only tell you: Yes, it is worth the effort. Even if you still have to read a lot yourself. And in the end you also have the bonus of having done pioneering work and being able to return a small contribution to the Digital Humanities community.

Translated with www.DeepL.com/Translator (free version)

Diesen Artikel zitieren / how to cite: Mareike K Schumacher, "MACHINE LEARNING: WHAT TO CONSIDER WHEN CREATING TRAINING DATA (FOR LITERARY STUDIES)," in *Lebe lieber literarisch*, March 31, 2020,

https://lebelieberliterarisch.de/en/machine-learning-what-to-consider-when-creating-training-data-for-literary-studies/, [zuletzt geprüft / Access: June 30, 2022].

References

Benz, N. (2020) *Ärgernis und Erkenntnis in der Named Entitiy Recognition*, *DH-Challenge*. Available at: http://dhchallenge.mareikeschumacher.de/argernis-und-erkenntnis-in-der-named-entitiy-recognition/(öffnet in neuem Tab) (Accessed: 5 February 2020).

Finkel, J. R., Grenager, T. and Manning, C. (2005) 'Incorporating Non-local Information into Information Extraction Systems by Gibbs Sampling', *Proceedings of the 43nd Annual Meeting of the Association for Computational Linguistics*, pp. 363–370. Available at: https://nlp.stanford.edu/~manning/papers/gibbscrf3.pdf.

Jannidis, F. *et al.* (2015) 'Automatische Erkennung von Figuren in deutschsprachigen Romanen ', *DHd 2015 Book of Abstracts*, pp. 2–6. Available at: http://gams.uni-graz.at/o:dhd2015.abstracts-vortraege.

Schumacher, M. (2020) *Automatische Erkennung von Figuren-Gender – das erste Modell*, *m*w*. Available at: https://msternchenw.de/automatische-erkennung-von-figuren-gender-das-erste-modell/ (Accessed: 5 February 2020).