Learning Machine Learning: A CRETA-Hackatorial on Reflected Text Analysis

Nils Reiter, Gerhard Kremer, Sarah Schulz

Introduction

The aim of this tutorial is to give the participants concrete and practical insights into a standard case of automatic text analysis. Using the example of automatic recognition of entity references, we will discuss general assumptions, procedures, and methodological standards in machine learning. The participants can fathom and test the scope of such procedures when editing executable programming code.

There is no reason to blindly trust the results of machine learning tools in general and NLP tools in particular. The concrete insights into the “engine room” of machine learning methods allow participants to more realistically assess the potential and limitations of supervised text analysis tools. Perspective, we hope to avoid the recurrent frustrations of using automatic text analysis techniques and their sometimes less than satisfactory results, and thus to promote the use and interpretation of the results of machine learning models. For their adequate usage in hermeneutic interpretation steps, the insight into influential technical details is indispensable. In particular, the type and origin of the training data is of importance for the quality of the machine-annotated data, as we will make clear in the tutorial.

In addition to a Python program for automatic annotation of entity references, with which we will work during the tutorial, we provide a heterogeneous, manually annotated corpus as well as the routines for evaluating and comparing annotations.

Entity References

The notion “entity reference” is deliberately broad and thus compatible with various research questions from Humanities and Social Sciences disciplines. In this way, different perspectives on entities can be considered. In total, five different entity classes are annotated in the corpus: PER (persons/characters), LOC (places), ORG (organizations), EVT (events) and WRK (works).

Entity references are phrases that refer to an entity in the real or a fictional world. Concretely, these are proper names (named entities, for example “Peter”), and appellative noun phrases (e.g., “the farmer”), as far as these refer to a concrete instance of the kind. The reference expression is always annotated as the maximum noun phrase (including article and prepositional phrases). Pronominal entity references are not annotated.
In **literary texts** character and location entities are of particular interest as basic categories of the narrated world. Character constellations and relations can be visualized on the basis of annotated entity references. At latest since the *spatial turn*, space has also come into focus as a relevant entity of the narrated world. In the **Social Sciences**, political parties and international organizations have always been central objects of analysis in empirical social research. The annotation of the ORG, PER and LOC entities in larger text corpora allows for multiple follow-up examinations, including on the visibility or evaluation of specific instances, such as the European Union.

**Text Corpus**

The basis for (supervised) machine learning methods are annotations. To automate the annotation of entity references, text data that covers the diversity of the entity concept is required. In the context of this tutorial, we will refer to annotations that have been created within the Center for Reflected Text Analytics (CRETA) at the University of Stuttgart (cf. Blessing et al., 2017; Reiter et al., 2017a).\(^1\) The corpus used in this tutorial contains one German literary text, one English literary text, and one German parliamentary debate.

**Johann Wolfgang von Goethe’s *The Sorrows of Young Werther*** is an epistolary novel from 1774. Our annotations are based on a revised version of 1787 and include the introductory words of the fictional editor as well as the first letters of Werther to his friend Wilhelm.

The **captivity narrative corpus** consists of eighteenth-century narratives which speak of the experiences of white settlers in North America who have fallen into native peoples’ captivity. Seven texts covering a total of 71,526 words were annotated using entity categories.\(^2\)

The **plenary debates corpus of the German parliament** consists of the plenary debates of the Bundestag recorded by stenographers and comprises 1,226 meetings between 1996 and 2015.\(^3\) Our annotations are limited to excerpts from a total of four plenary protocols which debated the contents of the European Parliament Union treat. Here, the entire speech of a politician was annotated for each protocol.

---

\(^1\)http://www.creta.uni-stuttgart.de

\(^2\)Made available under Creative Common license by EEBO-TCP which is a partnership between the Universities of Michigan and Oxford and the publisher ProQuest to create accurately transcribed and encoded texts based on the image sets published by ProQuest via their Early English Books Online (EEBO) database (http://eebo.chadwyck.com).

\(^3\)The texts were made available as part of the PolMine project: http://polmine.sowi.uni-due.de/polmine/
Procedure

The tutorial is based on so-called shared tasks from computational linguistics, but competitive aspect has mainly playful character in the tutorial. In a traditional shared task, the participating teams work on solutions to a single task often based on the same data. Such a defined task may be part-of-speech-tagging. Performing a simultaneous evaluation on the same gold standard, the developed systems can be directly compared. In our tutorial we play out this concept live and on site.

First, we briefly discuss the underlying texts and their annotation. Annotation guidelines are provided to the participants in advance. As part of the introduction, the specific organization of the annotation process will also be discussed so that the tutorial can serve as a blueprint for future activities of the participants in this and similar fields of work.

Participants then independently try to find a combination of machine learning algorithm, feature set, and parameter setting which predicts results that are most similar to the manually annotated gold standard of a different dataset. Specifically, this means that the impact of considered features (e.g., casing or length of a word) on the recognition of entity references can be tested empirically. Intuitions about the data and the annotated phenomenon are helpful here, as there is not enough time to simply test all possible combinations (“brute force”).

We consciously work without a graphical user interface (cf. Reiter et al., 2017b) – instead, participants edit the (Python) program directly after an introduction and under guidance. Prior knowledge in Python is not necessary: The program provided by us is designed so that even Python newcomers quickly understand the parts to be edited and used. If participants already have experience in Python programming, they can use advanced functionalities.

As usual at the end of machine learning processes, a performance evaluation will be carried out. For this purpose, the participants are provided with the final, previously unknown test corpus after a period of experimentation. The created models are applied to this data set, the resulting annotations are then compared to the gold standard and evaluated using standard metrics.

Learning goals

Using the example of automatic annotation of entity references, we demonstrate the steps required to automate a text analysis task using machine learning techniques and how they can be implemented concretely. The participants get a coherent overview: from the manual annotation of selected texts via the fine tuning of the learning process through to the evaluation of the results. This process is transferable to similar projects.

The tutorial improves the understanding of the relationship between the concept
under investigation and relevant features that are incorporated into a machine learning process. By gaining insight into the technical implementation, participants gain an understanding of the limitations and possibilities of automation in order to be able to realistically assess the potential of such approaches for their own projects. In addition, they are enabled to adequately critique results obtained on the basis of such procedures.

References


