# 8 Assumptions

## 8.1 Novel Assumptions

Two of the core assumptions that have been specifically relevant for modeling our approach have already been mentioned in the Introduction.

One of this is that *"Structure matters"* and that, because of this, the structure of input texts should be given more attention than in previous approaches. This concerns both implicit linguistic structure and explicit or implicit markup and formatting information. Our approach to taking this assumption into account is by modeling input texts as trees and the context of individual tokens as "inverted subtrees", instead of just considered text as a sequence of words as is usual in other approaches. More details on this will be given in Chap. 12.

Actually, this is more a conjecture than an assumption, since we will evaluate the effects of using such additional structural information in our system instead of just postulating that there will be a positive effect (Sec. 18.1).

The other mentioned assumption is that *"Systems will be used"* and that actual usage will typically involve a semi-automatic, interactive training regimen. For most "real-life" applications, automatic extractions will be checked and corrected by a human revisor, since automatically extracted data will always contain errors and gaps that can be detected by human judgment only. This correction process continually provides additional training data. However, typical trainable IE systems only support batch training from a set of annotated training texts. This makes them unsuited to integrate new data, since full retraining takes a long time.

To address this issue, our approach supports *incremental training* as an alternative to batch training, allowing successive refinement of an existing statistical model by dynamically adapting it to new training data. We will return to this when discussing which classification algorithms are suitable for our approach (Chap. 11).

## 8.2 General Assumptions

An assumption shared by all IE approaches is that textual documents may contain lots of information, most of which we are not interested in—or, at least, we might be interested in it but we will not be able to make use of it in structured queries. Even if it was possible to extract "all" relevant information from a text, we could not query it since we cannot know the resulting structure beforehand. We assume that the *target schema is predefined,* i.e., the kind of information to be extracted is specified before the extraction process starts. The kinds of target schemas which our system can handle will be discussed in the next chapter (Sec. 9.1).

With this limitation to predefined target schemas, we can also assume that no text understanding is necessary to extract the information we are interested in. One of the main criteria for text understanding is the ability to answer arbitrary questions to a text whereas IE "answers" only a fixed set of "questions" reflecting the target schema. This assumption justifies the use of machine learning models that can be trained on the training examples provided by humans without expensive background knowledge sources. Without this assumption, the task would be infeasible, since "understanding" in any usual sense of the term is outside the capabilities of current (at least) computers.

There are several other assumptions that are generally shared in the field of IE, but are seldom mentioned explicitly.

One of them is *corpus homogeneity*: Since the properties of the relevant extracted information have to be learned from training examples, training corpora should be sufficiently homogeneous, that is the texts in a training corpus are supposed to be similar in expression of relevant information. Ideally, training and application/evaluation corpus are random subsets of a full corpus, i.e., we have a set of documents to extract information from (the full corpus) and randomly draw a subset of documents to annotate and use for training. The remaining documents (or a random subset of them) are used for application or evaluation.

Actual applications will often deviate from this ideal model, e.g., documents for extraction will still be added after the training model has been built. This should generally be acceptable as long as the new documents are sufficiently similar to the old ones, but becomes problematic if the nature of the corpus changes over time. Incremental training takes such changes into account by allowing to gradually adapt the existing extraction model by training it on new documents (while batch training cannot integrate new training data without discarding the existing extraction model and rebuilding it from scratch). Hence incremental training reduces the effort of adapting the model to changes in the corpus. Also, since later training operations can overrule the effects of earlier operations, incrementally trained models will generally reflect more recent (later trained) documents more accurately than older documents, which is a good thing if the corpus changes over time. Even so, we have to assume that changes will be gradual—it would not make sense to apply an extraction model (whether trained incrementally or batched) to texts that differ radically from the training samples.

Supervised trainable annotations rely on annotated training data to build an extraction model. If the training data is inconsistent or erroneous, it might be impossible to build a consistent extraction model. Because of this, we have to rely on the *consistency and correctness of training data*. Occasional violations of this assumption will not cause the system to break down, but results are likely to suffer.

## 8.3 Suitability of Tasks

It is illusive to assume that the current approaches to IE will achieve comparable results for every kind of text. Obviously it is easier to find information in at least loosely structured form-like texts than in newspaper articles or even novels. The more variable and diverse a language is, the more difficult it is to determine common properties of extracted content. We can call texts comprising a certain regularity in structure and expression *technical texts*. We assume that the technical languages feature a restricted scope of expression possibilities for information and are therefore particularly suitable for information extraction. Examples of technical language are medical reports, economic news articles, regular announcements that tend to be expressed in similar ways (e.g. of seminars) etc.

The suitability of texts probably depends from several factors, among them the regularity and standardization of the terminology and expressions used as well as the style and degree of formality of the used language. A detailed analysis of these factors is an important research question, but beyond the scope of this work—we just have to assume that texts are reasonably suitable for IE.

Similarly, *facts to extract must be suitable* for IE. Specific and concise pieces of information (e.g. names or dates) are better suited for extraction than vague or loosely defined pieces of information (e.g. a “description”). The suitability of an attribute to extract will depend on various factors, among them the homogeneity or heterogeneity of possible values, their length, the typical placement in input texts. Again, we will not analyze these factors in detail; we just have to assume that they are sufficiently suitable and to accept that results will suffer if this assumption is violated.

Also, for modeling information extraction as a token classification task as described in Chap. 10 we have to assume that attribute values are *localized*, i.e., that each attribute value is expressed by a single, continuous text fragment. Each word is also assumed to be part of at most one attribute value—nested or overlapping attribute values are not supported.