Automation, Management and Improvement of Text Corpus Production

Christoph Kuras, Thomas Eckart, Uwe Quasthoff, Dirk Goldhahn
University of Leipzig
Augustusplatz 10, 04109 Leipzig
{ckuras,teckart,quasthoff,dgoldhahn}@informatik.uni-leipzig.de

Abstract

The process of creating large text corpora for different languages, genres, and purposes from data available on the Web involves many different tools, configurations, and – sometimes – complex distributed hardware setups. This results in increasingly complex processes with a variety of potential configurations and error sources for each involved tool. In the field of commercial management, Business Process Management (BPM) is used successfully to cope with similar complex workflows in a multi-actor environment. Like enterprises, research environments are facing a gap between the IT and other departments that needs to be bridged and also have to adapt to new research questions quickly. In this paper we demonstrate the usefulness of applying these approved strategies and tools to the field of linguistic resource creation and management. For this purpose an established workflow for the creation of Web corpora was adapted and integrated into a popular BPM tool and the immediate benefits for fault detection, quality management and support of distinct roles in the generation process are explained.

Keywords: corpus creation, process management, scientific workflows, BPM

1. Challenges of large-scale Text Corpus Production

Creating large text corpora for many different languages involves executing an extensive set of applications in a – more or less – defined order. This includes applications for preprocessing and annotation starting from sentence segmentation through to various annotation tools like part-of-speech taggers or parsers. For different kinds of text material and different languages there are typically varying configurations for each of these applications. Furthermore, the selection of applied tools might differ depending on the input material's language or language family as it is the case for special forms of word tokenization approaches. All in all this results in complex chains of tools with a variety of possible configurations.

The resulting solution has to be seen in the context of conflicting requirements: a systematic corpus production process has to be streamlined and automated in order to keep up with ongoing data acquisition, which may - in extreme cases - comprise minute-wise updates for news material or content obtained from social networking services. On the other hand, in collaboration with other researchers, new research questions arise continuously, making it crucial to be as flexible as possible when it comes to adaptions in the workflow. Another relevant aspect is the ability to trace errors in the running workflow. When executing a complex chain of applications, identifying errors in any of the processing steps can be time consuming especially if (partial) results are examined manually. As a consequence a systematic approach is needed to document configurations for every single execution of each application. Combined with an automatic data sampling many kinds of problems might be recognized, so processes can be interrupted already in an early stage to take any actions necessary. A thorough documentation of applied criteria also ensures the reproducibility of results; additionally the data can be used in terms of fault tracing.

Even more problem areas evolve when multiple persons or organizational units are involved in the creation process. This is for example the case when computing power is outsourced to commercial companies or when an external group of experts is in charge of reviewing or annotating data resulting from one of the processing steps. These external dependencies result in an even more complex process. This may lead – if not controlled and monitored properly – to inefficiencies due to disruptions in the process flow and becomes even more important when resources are processed in parallel. As a result, there is a demand for a system controlling the overall process execution and monitoring specific metrics which can be used to forecast execution time, allow statements about error rates at different steps, and similar issues.

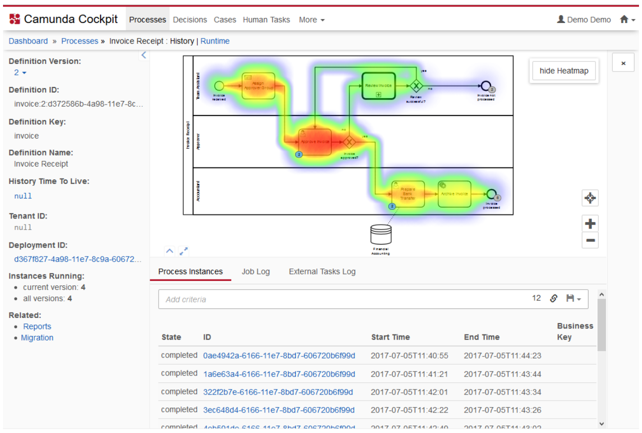
All these aspects are motivations for modeling and executing scientific workflows in Natural Language Processing (NLP). The existence of a variety of approaches, reaching from NLP-related tools like GATE through data analysis software like RapidMiner to supporting tools for managing scientific workflows like Apache Taverna or Kepler[1] illustrates the pressing demand in this area. In fact, a process-oriented view supported by powerful applications is already present in the field of economics for a long time. Business Process Management (BPM) (Aalst et al., 2000; vom Brocke et al., 2010; Aalst et al., 2016; Hirzel et al., 2013) has become extremely popular in commercial contexts and many of its features make it also useful for NLP-related tasks in a complex NLP environment.

2. Managing Corpus Creation with a Workflow-Management System

There are many tools available that can be used to model processes and that even allow to execute them. However, some of them solely model a data-centered view[2], describing how the data should be transformed, often including technical aspects of the respective implementation. These

[1] Which rely on a very generic definition for the term "scientific workflow": "an executable representation of the steps required to generate results.".

[2] Like RapidMiner or the Konstanz Information Miner.



Dashboard » Processes » Invoice Receipt : History | Runtime

Definition Version: 2

Definition ID: invoice:2:d372586b-4a98-11e7-8c...

Definition Key: invoice

Definition Name: Invoice Receipt

History Time To Live: null

Tenant ID: null

Deployment ID: d367f827-4a98-11e7-8c9a-60672...

Instances Running:
- current version: 4
- all versions: 4

Related:
- Reports
- Migration

State	ID	Start Time	End Time	Business Key
completed	0ae4942a-6166-11e7-8bd7-606720b6f99d	2017-07-05T11:40:55	2017-07-05T11:44:23	
completed	1a6e63a4-6166-11e7-8bd7-606720b6f99d	2017-07-05T11:41:21	2017-07-05T11:43:44	
completed	322f2b7e-6166-11e7-8bd7-606720b6f99d	2017-07-05T11:42:01	2017-07-05T11:43:34	
completed	3ec648d4-6166-11e7-8bd7-606720b6f99d	2017-07-05T11:42:22	2017-07-05T11:43:26	
completed	4eb591de-6166-11e7-8bd7-606720b6f99d	2017-07-05T11:42:40	2017-07-05T11:43:02	

Figure 1: Process heatmap in *Camunda Cockpit*

tools mostly define a model that includes operators shipping with that specific software. Other approaches, like Apache Taverna, are able to orchestrate different types of scripts and web services that can be hosted at remote locations. However, these applications are not capable of modeling the aspect of collaboration between organizations arising from the integration of hardware at different organizational units or even human beings. Moreover, many parallels can be drawn between the support of IT regarding research questions and the support of IT in business scenarios (cf. Kuras and Eckart (2017)). For that reason it seems natural to apply some of the strategies originating in that field. One of these strategies is Business Process Management (BPM). This is a management approach which is mainly targeted at streamlining processes within an organization and making the business more flexible, making it able to adapt to changes in the market quickly. One of the standard ways to model processes in this field is the Business Process Model and Notation (BPMN) which is a standard of the Object Management Group[3], currently in version 2 released in 2011 (OMG (2011)). Modeling in BPMN has the advantage that these models can be enhanced by technical specifications making it possible to execute them directly within a workflow management system (cf. Gadatsch (2012)). Popular solutions include jBPM, Activiti and Camunda.

For managing the considered corpus building process, the Camunda workflow management system is used[4]. The reasons for this decision lie especially in its open availability[5], its utilization in a variety of – often commercial – scenarios, and the availability of different helpful extensions and user-friendly interfaces. However, the use of BPMN as primary means of describing and executing workflows is not bound to a specific workflow management system; other software solutions could have been used instead.

Figure 1 shows a screenshot of the web interface to the Camunda process execution engine[6]. The system monitors the runtimes of each process instance and each task within this execution. This enables to generate a heatmap overlay revealing possible bottlenecks in the process by marking the tasks in which the process spends most of the execution time in average (red). This is a functionality that can be used only by monitoring the runtimes which is done by default. When modeling a process in Natural Language Processing, many different measures, which are called Key Performance Indicators (KPI) in the field of BPM, can be imagined (cf. Gadatsch (2012)). These measures can not only be used to monitor, streamline and improve the process itself but to ensure the quality of data being generated during the runtime of the process (see Section 4.).

3. Distributed Corpus Creation at the LCC

The Leipzig Corpora Collection (LCC) (Goldhahn et al., 2012) continuously generates corpora for a large number of languages. As more and more text material becomes available through the Web, massively-parallel crawling can result in amounts of raw data in the range of hundreds of gigabytes[7] that have the potential to pile up to almost unprocessable "data heaps".

To handle these amounts of data in an acceptable period of time the already established processing workflow was extended by integrating an external computing center. This computing center provides a high-performance computing (HPC) cluster via a RESTful API using the UNICORE interface (Uniform Interface to Computing Resources[8]). The overhead of delegating working steps to an external computing facility consists here mostly of data transfer times and is in the current configuration – at least compared to the actual data processing – rather slim.

Figure 2 depicts the coarse model of the process in BPMN notation. In the first step, available raw data is selected, which then gets preprocessed using the resources of the external partner. After that, the data is enriched locally by the calculation of cooccurrences and finally imported into a relational database.

An important aspect of BPMN is the possibility to model subprocesses hiding complexity. On one hand this enables personnel not familiar with the process to quickly get an overview. On the other hand, expected faults of the process execution can be modeled directly. This enables the process engine to decide which actions have to be taken in case of an error, making the execution even more efficient by saving the costs of additional human interactions. Figure 3 shows a more detailed variant of the preprocessing subprocess. It basically consists of these steps:

- *sentence segmentation*: segment raw text data into sentences
- *sentence cleaning*: remove sentences that are, based on patterns, undesirable

[3] `https://www.omg.org`

[4] `https://camunda.org`

[5] The Camunda platform is licensed under the Apache License 2.0.

[6] `https://docs.camunda.org/manual/7.7/webapps/cockpit/bpmn/process-history-views/`

[7] 4 Terabytes of incoming raw material per day are typical values for LCC crawling processes.

[8] `https://www.unicore.eu/`

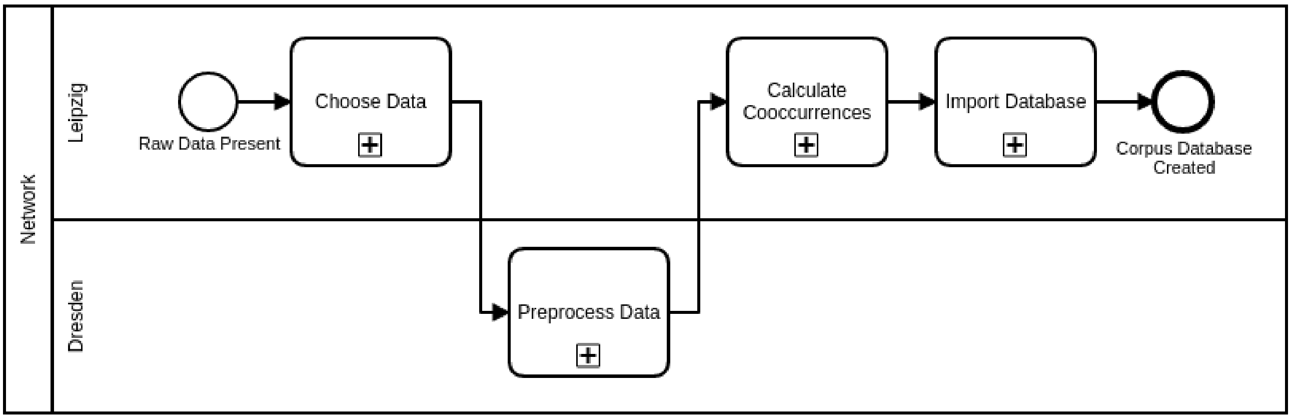


Figure 2: Coarse model of the corpus production steps

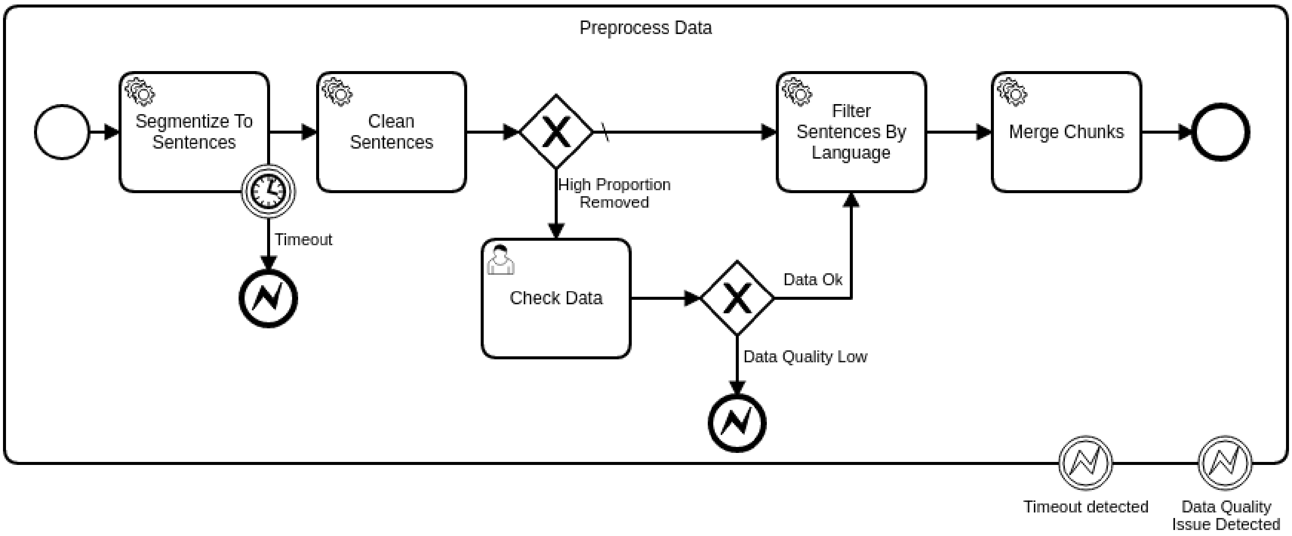


Figure 3: More elaborate model of the preprocessing subprocess

- *language filtering*: filter out sentences not belonging to the target language
- *merging*: merge all chunks of sentences (due to parallel processing)

To illustrate the possibilities of BPMN modeling the subprocess is enhanced by different mechanisms to detect fault during the process execution. An *intermediate boundary timer event* is added to the *Segmentize To Sentences* task. If the task exceeds a defined time limit, an *error end event* will be thrown. On the subprocess border, a *boundary error catching event* for the error kind “Timeout” is installed. This will ensure that all thrown “Timeout” events from within the subprocess will be caught and can then be handled outside the subprocess. After cleaning the sentences by removing lines matching predefined patterns (e.g. sentences containing too many digits or that are excessively long), a decision with respect to the sequence flow is modeled. Based on the available process data, the execution engine automatically selects the path to execute by evaluating a predefined condition[9]. At this point, the process engine will choose the path depending on the proportion of removed sentences. The decision is based on the measuring data available during execution; the exact calculation has to be specified in the model. In case the proportion is above a predefined threshold, a *User Task* will be assigned to a human actor who has access to the Camunda web interface. The user is automatically informed about the assignment of the new task; it is also possible to assign a task to a group of authorized users from which one person can then claim the task. A small set of randomly sampled sentences can be presented to the user who will then decide whether there is a general issue with the quality of the underlying data. According to the user decision, the process will throw a data quality related error or continue the process execution normally. The actual error handling may include a message to personnel responsible for the input data.

4. Measurement and Improvement of Corpus Quality

One of the main purposes of BPM is the continuous improvement and quality assurance of the managed processes (cf. Reichert and Lohrmann (2010)). This is also an important aspect concerning the production of high quality lin-

[9] The decision is modeled using a XOR-Gateway.

guistic resources. The area of fault detection and quality assurance can be seen and handled with a focus on a variety of criteria. This includes setting the evaluation focus on the process itself or on the actual results of every intermediate step and the final results. Both require a systematic monitoring and appropriate procedures if deficits were identified. Typical criteria for the first evaluation – focused on rather "technical" indicators – include, for example, basic information if processes or subprocesses terminated unexpectedly, the extent of technical resources that were consumed, or technical constraints that were or were not met (including weak or strong temporal constraints).

A key aim of such a thoroughly monitored environment is the extraction and identification of process patterns. Over time, expectancy values for all metrics evolve so that a problematic process instance can be revealed at the earliest possible moment[10]. Based on a set of predefined rules this allows the workflow management system to decide automatically whether to cancel the process, saving processing time and resources. Furthermore, these metrics can be used to make statements about the quality of the overall outcome right after the processing has finished, for example how "noisy" a corpus is or how it compares with similar resources. This does not only apply to the process as a whole but to each single task involved in the process, making it easier and faster to spot problems during execution, which is especially important when executions run for long periods of time. Manually performed checks, though feasible with smaller data and less frequent executions, would be error-prone and inefficient with respect to an "industrial-scale" corpus production that is done within the context of the LCC. Supporting and controlling processes with a workflow management system ensures the systematic application and evaluation of all relevant criteria.

These criteria are a typical starting point for the identification of general problems or performance issues like the identification of bottlenecks, implementation inefficiencies or alike. As the processing of big data material often has to deal with performance issues and the efficient usage of available hardware, optimizing the structure of those processes is of special interest. This also requires a detailed recording of the processes' tasks runtimes and latency times for a larger – representative – number of executions. Parallels to other disciplines are therefore hardly surprising: many standard metrics in the field of logistics apply for the "logistics" of NLP pipelines as well and can be used as inspiration, e.g. analysing historical data being able to forecast trends concerning future resource needs like storage and processing power (cf. Robinson (1989)). Data storage can be seen as a stock, especially when multiple storages are needed due to the integration of remote computing centers and the data needs to be transferred from one stock to another[11]. Avoiding supply shortages as well as excess stocks, e.g. as a result of an imbalance between data collection and processing time, is crucial for an efficient use of resources during the processing of large corpora. The creation of text corpora is an end-to-end process that reaches from the collection of raw data through to the delivery to an end user, thus can also be considered as a supply chain that involves many suppliers of data, providers of services and end users demanding the actual outcome of the process.

A less technical viewpoint focuses on the actual outcomes of a process, which is data as the result of a sequence of subprocesses. Quality of data is typically related to its usability for a specific purpose and hence, often hard to measure automatically. However, even simple and easily determinable criteria may function as useful indicators. In the case of NLP tools this may be the comparison of input material to output material sizes (like the amount of raw text with the resulting number of sentences, types, or tokens), checking the completeness of different annotation layers, or identifying untypical distributions of annotations for the language family, language, source or genre in question. Additionally, any linguistic invariant may function as a "deep" indicator for the well-formedness of language material, especially in cases where input data is of uncertain quality. As this often requires specific annotations or even human intuition, simple principles based on language statistics may sometimes suffice as a substitute (Eckart et al., 2012). In any case, BPM allows their integration and more checks as an integral component of the execution and evaluation of every process instance. Furthermore, it provides build-in capabilities to support both automatic and manual checks. These data not only can increase the quality of the outcome of a single process instance based on automatic sequence-flow decisions. It can also build the foundation for the analysis of historical process data allowing assertions about the performance of subprocesses and the consumption of resources. For instance, recorded process data in a test case revealed the proportions of average subprocess time consumptions to be 34% for preprocessing, 64% for the calculation of cooccurrences and 1% for the database creation. As more data are collected, more sophisticated statements about the impact of text type and data size on these measures are possible. Historical data can also be used to spot configuration problems concerning specific languages. Regarding the loss of size after filtering the sentences by the target language, a size reduction ranging from 3% to 18% was observed in a test case, depending on language and origin of the data. However, such deviations can also indicate configuration problems. Another measurable aspect is the throughput of data of the involved services which may also reveal patterns pointing towards configuration problems or quality issues with resources used by these services. Furthermore, this allows forecasting of runtimes, again with respect to language and origin of the data, making the prediction more reliable. As some checks require profound knowledge in specific fields, e.g. linguistics, manual checks might still be necessary. With an appropriate process model and a workflow management system, even such manual checks, often completely isolated from fully-automated tasks, can be integrated into the process execution. Inspections by domain or language experts may function as prerequisite for the publication of a resource. Comparable to other workflows in a highly specialized working environment this requires a consistent rights

[10] Those values are of course specific for different characteristics of the input material like its language, source format or origin.

[11] In such computing centers, high performance data storage can be strictly limited.

management and the assignment of accurate process roles.

5. Summary

In contrast to many "proprietary" solutions that are often used in practice, BPMN is a well known and documented standard that supports a common understanding of processes, interfaces and interrelations for all involved participants. It can be used as a standardized description of a workflow and is at the same time executable in different workflow management solutions. Being an established standard, it has the benefit of support by different software tools and simplifies reuse in other contexts.

Both the techniques and the tools used in Business Process Management prove to be useful in the process of the repeated production of large corpora. BPM allows the definition of complex processes using a heterogeneous infrastructure for different corpus processing tools and intermediate human interaction. This allows an embedded quality control for the data which are processed. In the case of adaptations of the corpus creation process (for instance, with a special word tokenization tool for a new language), this can be modeled transparently. Furthermore, the approach is highly flexible as it allows reusing tasks in new process models or extending fully automated processes by human interaction without having to modify task implementations. In addition to that, replacing underlying implementations of tasks is possible without changing the process itself. The model is non-technical and understandable for non-programmers; parameters like runtime distribution can be visualized user-friendly for process monitoring.

6. Bibliographical References

Aalst, W. v. d., Desel, J., and Oberweis, A. (2000). *Business Process Management Models, Techniques, and Empirical Studies*. Lecture Notes in Computer Science. Springer-Verlag, Berlin, Heidelberg.

Aalst, W. v. d., La Rosa, M., and Santoro, F. M. (2016). Business process management. *Business & Information Systems Engineering*, 58(1):1–6, Feb.

Eckart, T., Quasthoff, U., and Goldhahn, D. (2012). Language Statistics-Based Quality Assurance for Large Corpora. In *Proceedings of Asia Pacific Corpus Linguistics Conference 2012, Auckland, New Zealand*.

Gadatsch, A. (2012). *Grundkurs Geschäftsprozess-Management - Methoden und Werkzeuge für die IT-Praxis: Eine Einführung für Studenten und Praktiker*. Springer Vieweg, 7. edition.

Goldhahn, D., Eckart, T., and Quasthoff, U. (2012). Building Large Monolingual Dictionaries at the Leipzig Corpora Collection: From 100 to 200 Languages. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation*, pages 759–765.

Hirzel, M., Geiser, U., and Gaida, I. (2013). *Prozessmanagement in der Praxis - Wertschöpfungsketten planen, optimieren und erfolgreich steuern*. Springer Gabler, 3. edition.

Kuras, C. and Eckart, T. (2017). Prozessmodellierung mittels BPMN in Forschungsinfrastrukturen der Digital Humanities. In *INFORMATIK 2017, Lecture Notes in Informatics (LNI)*.

OMG. (2011). Business process model and notation BPMN - Version 2.0. `http://www.omg.org/spec/BPMN/2.0/PDF`, retrieved 2017-09-28.

Reichert, M. and Lohrmann, M. J. (2010). *Basic considerations on business process quality*. Ulmer Informatik-Berichte. Universität Ulm. Fakultät für Ingenieurwissenschaften und Informatik, Ulm.

Robinson, D. (1989). IT in Logistics. *Logistics Information Management*, 2(4):181–183.

vom Brocke, J., Rosemann, M., and Bernus, P. (2010). *Handbook on Business Process Management*. International Handbooks on Information Systems. Springer, Heidelberg u.a.