Discovering Structural Similarities in Narrative Texts using Event Alignment Algorithms

Dissertation zur Erlangung der Doktorwürde
im Fach Computerlinguistik
der Neuphilologischen Fakultät
der Ruprecht-Karls-Universität Heidelberg

vorgelegt von

Nils Reiter

Publication	June 2014
Commission chair	Prof. Dr. Ekkehard Felder Germanistisches Seminar Universität Heidelberg
Supervisor and first reviewer	Prof. Dr. Anette Frank Institut für Computerlinguistik Universität Heidelberg
Secondary reviewer	Prof. Dr. Sebastian Padó Institut für Maschinelle Sprachverarbeitung Universität Stuttgart

Acknowledgements

I would like to thank my supervisor, Anette Frank, for supervising and guiding this thesis. In particular, I am thankful for the opportunity to delve into details, for the continuous support, the engaging discussions and the critical yet fruitful feedback. It was a pleasure to work in and with her entire group. In particular, I would like to thank my year-long office mate Eva Sourjiková for the pleasant office atmosphere, my friend and colleague Michael Roth for all the fun we had and Matthias Hartung for . . . I don't know where to start.

Furthermore, I thank "my" ritual experts, Oliver Hellwig and Christof Zotter, for answering questions and doing annotation work. I also thank Thomas Bögel, Irina Gossmann, Mareike Hartmann, Borayin Larios, Julio Cezar Rodrigues and Britta Zeller for working in this ritual research project and doing implementation and annotation. All this would not have been possible without the Sonderforschungsbereich 619 and the funding by the German Research Foundation, for which I am equally thankful.

Following Propp, folktales have magical helper agents, and so does the tale of working on this thesis. I am very thankful to my helper agents and for the encouragement they gave me.

Nils Reiter

Abstract

This thesis is about the discovery of structural similarities across narrative texts. We will describe a method that is based on event alignments created automatically on automatically preprocessed texts. This opens up a path to large-scale empirical research on structural similarities across texts.

Structural similarities are of interest for many areas in the humanities and social sciences. We will focus on folkloristics and research of rituals as application scenarios. Folkloristics researches *folktales*, i.e., tales that have been passed down orally for a long time. Similarities across different folktales have been observed, both at the level of individual events (being abandoned in the woods) or participants (the gingerbread house) and structurally: Events do not happen at random, but in a certain order. *Rituals* are an omnipresent part of human behavior and are studied in ethnology, social sciences and history. Similarities across types of rituals have been observed and sparked a discussion about structural principles that govern the combination of individual ritual elements to rituals.

As descriptions of rituals feature a lot of uncommon language constructions, we will also discuss methods of domain adaptation in order to adapt existing NLP components to the domain of rituals. We will mainly use supervised methods and employ retraining as a means for adaptation. This presupposes annotating small amounts of domain data. We will be discussing the following linguistic levels: Part of speech, chunking, dependency parsing, word sense disambiguation, semantic role labeling and coreference resolution. On all levels, we have achieved improvements. We will also describe how these annotation levels are brought together in a single, integrated discourse representation that is the basis for further experiments.

In order to discover structural similarities, we employ three different alignment algorithms and use them to align semantically similar events. Sequence alignment (Needleman-Wunsch) is a classic algorithm with limited capabilities. A graph-based event alignment system that has been developed for newspaper texts will be used in comparison. As a third algorithm, we employ Bayesian model merging, which induces a hidden Markov model, from which we extract an alignment. We will evaluate the algorithms in two experiments. In the first experiment, we evaluate against a gold standard of aligned descriptions of rituals. Bayesian model merging and predicate alignment achieve the best results, measured using the Blanc metric. Due to difficulties in creating an event alignment gold standard, the second experiment is based on cluster induction. Although this is not a strict evaluation of structural similarities, it gives some insight into the behavior of the algorithms.

We induce a document similarity measure from the generated alignments and use this measure to cluster the documents. The clustering is then compared against a

gold standard classification of documents from both scenarios. In this experiment, the lemma alignment baseline achieves the best numerical performance on folktales (but as it aligns lemmas instead of event representations, its expressiveness is limited), followed by predicate alignment, Needleman-Wunsch and Bayesian model merging. On descriptions of rituals, the predicate alignment algorithm outperforms all baselines and the other algorithms. Shallow measures of semantic similarities of texts outperform the alignment-based algorithms on folktales, but they do not allow the exact localization of similarities.

Finally, we present a graph-based algorithm that ranks events according to their participation in structurally similar regions across documents. This allows us to direct researchers from humanities to interesting cases, which are worth manual inspection. Because in digital humanities scenarios, the accessibility of results to researchers from humanities is of utmost importance, we close the thesis with a showcase scenario in which we analyze descriptions of rituals using the alignment, clustering and event ranking algorithms we have described before. We will show in this showcase how results can be visualized and interpreted by researchers of rituals.

Contents

List of Tables

List of Figures

1 Introduction

Narrative texts are important textual sources in the humanities and social sciences. In particular, many research questions revolve around similarities, parallels or overlaps across narrative texts, e.g., discovery of intersections in biographies or similar developments of characters in fictional tales. Within the emerging paradigm of digital humanities, we develop methods and a system for the automatic discovery of story similarities in narrative texts. The system uses advanced computational linguistics techniques and is designed for the specific needs and premisses of digital humanities.

We study two application scenarios in detail: Research of folktales and of rituals. In both areas, the detection of similarities across stories plays a major role. For folktales, the classification of tales from different eras and cultures based on common elements in their story lines has been studied since the early twentieth century. The discovery of reoccurring elements in rituals has sparked a discussion about the existence of structural principles that rule the combination of these elements. For these principles, the term "ritual grammar" has been coined[1].

Both application scenarios suggest quantitative-empirical research approaches as they aim at analyzing more than a single textual source. In addition, the expressiveness of the analysis increases the more sources have been studied. However, traditional research approaches prevalent in the humanities do not scale easily, as they rely on human scholars (close-)reading texts. A system that automatically detects similarities in narratives is a key component for enabling large-scale empirical research in this direction[2].

The system we describe detects story similarities using event alignment algorithms. The alignment algorithms work on densely connected discourse representations, which contain representations for events, characters and the linguistically analyzed textual source data. Using measures of semantic similarity for events, the alignment methods detect story similarities across discourses that can be visualized and thus made accessible to humanities researchers.

The discourse representations themselves are created fully automatically using state of the art techniques on a variety of different linguistic analysis levels. Given the peculiar text characteristics of the descriptions of rituals, the linguistic processing tools are adapted to the domain of rituals. Not all processing tools for the different linguistic annotation levels employ the same basic methodology. Consequently, the adaptation of these tools must employ different adaptation strategies. Supervised linguistic

[1] This thesis was written within the context of the research project "Ontology modeling for ritual structure research", in the collaborative research center "Ritual Dynamics", funded by the German Research Foundation (Sonderforschungsbereich 619, Ritualdynamik).

[2] In the words of Moretti (2000), we are describing a distant-reading approach.

processing methods suggest retraining as a simple yet robust adaptation strategy. For the adaptation of both the word sense disambiguation and the coreference resolution system we employed ways of incorporating domain knowledge. The unsupervised, knowledge-based word sense disambiguation system was adapted by enhancing the knowledge-base directly. In order to adapt a coreference resolution system, we employ multiple ways of integrating domain knowledge into the process.

For the detection of story similarities, we use three different alignment algorithms. The algorithms use a similar set of features for measuring the semantic similarity of events. We evaluate the three algorithms on a gold standard of descriptions of rituals and in a clustering-based evaluation. In the latter, the density of produced alignments is used as a measure for document similarity, which is in turn used by a clustering algorithm. In order to provide researchers from the humanities a means for targeted inspection, we also describe an algorithm for the discovery of dense alignment regions between narratives.

As a showcase scenario, we show how a researcher of rituals can inspect and interpret the data structures that we produce. This includes the induced document similarity, the underlying alignment links and individual densely connected regions. We describe how these can be interpreted and show possible visualizations for them.

Structure of the Thesis

The structure of this thesis is as follows. We will first give an introduction into the field of digital humanities in general (Chapter 2). We will discuss the role of computational linguistics in digital humanities, highlight the main challenges computational linguistics faces and discuss how they affect our work.

We will discuss related work to this thesis in Chapter 3. As the two main parts the thesis will be concerned with are on domain adaptation and narrative analysis, Chapter 3 is structured accordingly: Section 3.1 describes related work on domain adaptation for supervised NLP components. As word sense disambiguation is often performed using unsupervised methods, we will also investigate domain adaptation methods for unsupervised word sense disambiguation algorithms. Existing work on computational narrative analysis is discussed in Section 3.2, which also gives a background on narratives in general.

In Chapter 4 we will describe the two application scenarios in detail: Section 4.1 on folktales and Section 4.2 on rituals. For each scenario, we will give a short overview of the state of research and describe why similarities across narratives are important. We will also discuss how they can benefit from the computational analysis methods we employ. Finally, we will introduce the corpora we collected in order to conduct experiments.

Chapter 5 first describes the processing architecture, its in- and output and key characteristics. The major part of the chapter describes the domain adaptation strategies we employed for processing the descriptions of rituals on the following linguistic levels: part of speech tagging, chunking, dependency parsing, word sense disambiguation, semantic role labeling and coreference resolution. All these levels are integrated into the discourse representation.

Our methodology for the automatic discovery of story similarities is described in Chapter 6. We will first give a general overview of the methodology and the specific experiments we are conducting. The algorithms will be evaluated in two experiments using data from both application scenarios. We will also describe an algorithm for identifying the most dense alignment regions that are worth investigating.

In Chapter 7, we will describe how a researcher from humanities can make use of the analysis tools we can provide. We will also describe how alignment-based story similarity, the alignments themselves and dense alignment regions can be visualized and interpreted. Chapter 7 shows visualizations and analyses that can be performed on descriptions of rituals as a showcase. The analysis starts globally, by inspecting document similarities as a whole, and delves deeper in a stepwise fashion, from finding densely connected regions to analyzing individual structural similarities.

We will conclude the thesis in Chapter 8 with a summary of our solutions to the challenges computational linguistics faces within digital humanities and our scientific contributions. Finally, we will give an outlook on future work and other potential application scenarios.

2 Digital Humanities

Digital humanities is no clearly defined research area. In contrast, the term is used as an "umbrella term" (Presner and Johanson, 2009) that encompasses a wide variety of areas employing computational methods to answer or address research questions from various humanities disciplines. The umbrella covers not only classical humanities, but also social sciences, history, archeology and many others. Consequently, the computational methods employed cover a wide variety, and computational analysis of language and texts is but one of many. Other computational fields are image recognition, visualization and 3D modeling, in fields such as archeology or art history.

The work that is often reported to be the first in digital humanities (e.g. in McCarty, 2003), however, has a linguistic background: In 1946, Roberto Busa started creating the *Index Thomisticus*, a concordance of the writings of Thomas Aquinas, with some support of IBM (Busa, 1980). In its final stage, the corpus contained 11 million tagged and lemmatized tokens. A detailed description of the history of digital humanities can be found in McCarty (2003) and Raben (1991).

Given the fact that linguistics has a long tradition in the humanities, computational linguistics (CL) can (in part) be seen as a prototypical digital humanities discipline: Scientific study of language (linguistics) is carried out using computational methods and from a computational perspective. This led not only to novel theories about language (e.g. formal grammar theories), but also to novel computational methods specific for (or mainly used in) computational linguistics (e.g. parsing techniques) and illustrates how both the humanities and the computational discipline can benefit from such interdisciplinary work[1].

Apart from linguistics, many more humanities disciplines are using natural language and in particular texts as their main research object: Newspaper articles are studied in social and political sciences, poetry and prose is studied in literary science and old records, charters and documents are studied in history (to name a few). Language analysis methods are therefore of interest in many digital humanities disciplines, as they allow analyzing texts on a large scale or support uncovering quantitative text properties that are not directly accessible. Computational linguistics can play a central role in the language-oriented digital humanities areas and in fact, a number of research projects have been carried out that use methods, techniques or representations from computational linguistics.

[1] Clearly, this is not the only perspective on computational linguistics one can take.

	Reference	Linguistic levels	Text genre	∅# tokens	Goal
Literary science	Inaki and Okita (2006)	surface	novel	28, 019	Analysis of different roles of a character in different novels
	Clement (2008)	surface	novel	517, 207	Visualization for discovering structure in repetitions
	Elson et al. (2010)	NE, flat syntax	novel	~ 170, 000	Creating a social network of characters in narrative fiction
	Brooke et al. (2013)	lemmas, POS	poetry	3, 533	Clustering voices in a poem
History	Jockers et al. (2008)	surface	religious book	266, 963	Author attribution
	Camp and Bosch (2012)	lemmas, POS, NE	biography	887, 404	Extracting social relations from biographies
	Cybulska and Vossen (2011)	lemmas, POS, word senses	newspaper, encyclopedic texts	16, 932	Extracting historic events from texts

Table 2.1: Existing digital humanities research using methods from computational linguistics

2.1 Existing Computational Linguistics Research within Digital Humanities

Table 2.1 shows an overview of several research works from literary sciences and history. This is not a comprehensive list, but illustrates the wide variety and some of the challenges computational linguistics faces in digital humanities research. The table shows the linguistic representation levels used, the text genre, the average number of tokens[2] and a short description of the task.

A first observation is that the texts under study play different roles. In some projects, the text itself is the research object, while in others the text serves as a medium and the research object is the information contained in the text. In the former, researchers are often interested in stylistic aspects (e.g. beauty of poems, Kao and Jurafsky, 2012) or properties of the text as a whole (e.g. the author, Jockers et al., 2008). The latter can be seen as information extraction tasks, but with a humanities application scenario (e.g. social network extraction, Camp and Bosch, 2012). It also has to be noted that there is a gray area in between these two poles. In history, for instance, the interpretation of extracted information may depend on meta data of the source document (why it was written, by whom, etc.).

As a second observation, we note that the data set sizes are relatively small, in comparison to corpus sizes in general computational linguistics[3]. Given the fact the main benefit of using computational methods is being able to process large data sets, this is somewhat surprising. In some cases, this can be explained by a very focused application goal. There is no point in "using more data" if the goal is an analysis of a specific literary piece (e.g. in Clement, 2008; Inaki and Okita, 2006; Jockers et al., 2008). In other cases, analyzing more data makes sense on a conceptual level, but more data is not available currently and also will not be available in the future (e.g. Camp and Bosch, 2012).

Thirdly, most of the existing work in computational linguistics for digital humanities makes use of rather shallow linguistic representations, even in the information extraction tasks. Although approaches using shallow linguistic representations are popular in general computational linguistics as well, the lack of approaches using deep linguistic structures is striking. On the first sight, this is surprising, in particular given the small data set sizes. Deeper linguistic representations would allow more fine-grained and meaningful analyses. However, the automatic creation of deep linguistic representations is technically difficult, and in particular for texts from non standard domains also error-prone.

[2] The number shown in the table refers to the average number of tokens analyzed. In Elson et al. (2010), for instance, the total number of tokens is much higher, as the entire corpus contains 60 novels. As the analysis is done per novel, we show the average number of tokens per novel in the table. In Cybulska and Vossen (2011) and Inaki and Okita (2006), different parts of a larger corpus have been studied in comparison.

[3] The Wall Street Journal part of the Penn Treebank (Marcus et al., 1999) contains one million tokens , the latest release of the Gigaword corpus (Parker et al., 2011) contains four million tokens.

2.2 Challenges for Computational Linguistics

To synthesize, there are a number of challenges for the application of computational linguistic techniques to textual data from the humanities:

Corpus sizes Modern computational linguistic approaches often rely on huge text corpora and use sophisticated methods to train statistical models on the corpora. These approaches are difficult to apply directly, because most text corpora in humanities are relatively small and focused. Literary analyses, for instance, need to be based on a certain fixed data set, that also can not be expected to grow in time. Although corpus-based analyses of literary pieces have been made (e.g. Inaki and Okita, 2006), the relatively small data size imposes restrictions on the methodology.

Text characteristics The text corpora used in computational linguistics are often newspaper corpora. Owing to different text characteristics on a number of linguistic levels (e.g. lexicon, style, syntax), reusing existing models that have been trained on newspaper corpora often leads to unsatisfactory results. Existing models need to be adapted to the characteristics of the texts found in humanities. As there is not a single text genre in humanities (or even within literary analysis), it is doubtful whether a single allround-adaptation will suffice. Instead, adaptations need to be done specifically for genres and texts at hand. This adaptation is not only required for (computational) analysis tools, but for annotation guidelines (and subsequently linguistic theory) as well, because most guidelines make assumptions that may not be met by the domain at hand.

The computational analysis of historical texts, for instance, is severely challenged by the fact that spelling and grammar are very heterogeneous in historical documents (Dipper, 2011). Poetry texts, on the other hand, often feature a unified spelling, but a rather loose syntax. In addition, poetry uses verse as a level of information that is not even present in newspaper texts but still very relevant for the interpretation in literary science and thus needs to be represented computationally (Kao and Jurafsky, 2012). Corpora containing computer-mediated language (e.g. chat logs) again differ from historical texts and poetry, eg by the use of colloquial forms or emoticons (Beißwenger and Storrer, 2009).

Category definitions Many tasks and research questions in humanities are extremely complex and so are the categories that computational methods should reproduce. Formal definitions for these categories often do not exist. Similarly, systematic annotations using guidelines and measuring annotator agreement are very rare in humanities. This in turn makes evaluation and supervised statistical approaches difficult to apply, as there is no data to evaluate against or train on[4]. Evaluating quantitatively, however, is

[4] What poses a challenge for computational approaches can be seen as a chance for humanities. McCarty (2003) points out that the "inevitable" mismatch between informal categories existing in humanities and their formalizations "forces ontological questions that lead back to [. . .] fundamental problems" in the humanities discipline. Humanities disciplines are forced to rethink their category system, which

a key advantage of computational methods (and goes together with processing large(r) data sets).

As an example, consider the literary discussion about clichés in the "Eumaeus"-episode of James Joyce's *Ulysses*. It has been argued in literary science that clichés are used a lot in order to reproduce the everyday language of uneducated people of Dublin in 1904 (Byrnes, 2010). The number of clichés in this episode has been *estimated* to be high, but was never quantified exactly (and thus was not systematically comparable to other literary pieces). Byrnes (2010) published a study in which he manually *counted* the number of clichés, based on his language intuition as a native speaker, Google and dictionaries of idiom and cliché. However, a sound definition of the concept of a cliché and what distinguishes clichés from other kinds of idioms is lacking. In consequence, computational linguistics approaches in this direction (e.g. Cook and Hirst, 2013) are either forced to establish a new definition or circumvent this issue by not using training material and/or devising other ways of evaluation.

Accessibility Results produced with computational linguistics methods need to be accessible to (digital) humanities researchers. Results, in this case, do not only include tables with numerical performance measures. Instead, automatically induced annotations need to be presented along with the original source texts, such that the results are traceable (to a certain extent) and humanities researchers can base their interpretations on them. Original source texts may also need to be cited as evidence in publications.

Making results accessible to humanities researchers is a challenge that goes beyond pure engineering, though. Although some visualizations may be rather obvious (e.g. showing social networks as graphs), others highly depend on the research question at hand. Clement (2008), for instance, shows how visualizations of repetitions in Gertrude Stein's *The Making of Americans* can support certain interpretations in literary science. While the computational linguistics part of the work is straightforward, the concrete visualization is pertained to the specific question and a result of a collaboration of literary scholars with technicians and designers in the project.

Sculley and Pasanek (2008) go even further and state that assumptions, implicit biases and limitations made in computational (in this case: machine-learning) methods need to be *understood* by the humanities researchers. Computed results should not be seen as a proof or determinate answer and methods should not be treated as a black box. This requires close collaboration between computational experts and humanities researchers and goes beyond visualization of results. Instead, Sculley and Pasanek require scientists from humanities to acquire at least a basic understanding of statistical and computational methods.

2.2.1 Challenges for the Detection of Structural Similarities

The challenges discussed above also affect this work on various levels. The number of descriptions of rituals is, compared to corpus sizes in computational linguistics, rather

can be a fruitful process.

small. Although a huge amount of folktales do exist in principal, a well structured, machine-readable corpus is not available directly. Therefore, we have to cope with relatively small data set sizes. As we will see, the uncommon text characteristics of the descriptions of rituals play a major role and cause us to develop various domain adaptation techniques. Category definitions (structural similarities across rituals or folktales) do not exist in a formal, controlled fashion. This makes the annotation of a gold standard difficult. In order to have an additional evaluation that is not dependent on these annotations, we performed the second experiment that makes use of a classification of rituals and folktales. This classification is sufficiently formal (and consensual in the respective fields) to be used in our setting. Finally, in order to produce results that are usable and accessible to researchers from folkloristics and research of rituals, we developed a number of tools to allow visualization, targeted inspection and fine-grained analysis of the structural similarities we detect automatically.

2.3 Summary

Digital humanities is a growing field of research and encompasses many different disciplines. It is unclear whether a clearly defined set of methods will ever emerge as "digital humanities methods".

Many humanities areas are using texts as either research objects or medium. Therefore, methods from computational linguistics may be of great use in these disciplines. However, the application of computational linguistics methods in digital humanities scenarios poses a number of challenges. Solutions to these challenges need to be focused on the specific tasks and data at hand.

3 Related Work

The related work to this thesis falls into two general areas: Domain adaptation and computational narrative analysis. In Section 3.1, we will discuss approaches on domain adaptation that have been used for various linguistic levels of annotation. As most of the linguistic processing methods are probabilistic, supervised methods, most of the adaptation approaches that we will discuss focuses on these methods. In addition, we will discuss approaches for the adaptation of unsupervised knowledge-based word sense disambiguation. The related work to computational narrative analysis can also be separated in two general areas: Approaches for representing and modeling single narratives and approaches for the comparison and aggregation of multiple narratives. Section 3.2 is structured accordingly.

3.1 Domain Adaptation

This section discusses existing work in the area of domain adaptation in the following way: Domain adaptation techniques that are applicable to supervised approaches are discussed in Section 3.1.1. Approaches for domain adaptation of knowledge-based word sense disambiguation will be discussed in Section 3.1.2.

3.1.1 Domain Adaptation for Supervised Approaches

Supervised techniques work by inducing statistical models on training data and applying them to test or application data, which should be a different data set. It is assumed that both data sets are samples drawn from the same underlying distribution. If, however, the data sets come from different domains, this assumption does not hold.

In the following discussion, we will assume that there are two domains under study. The source domain is one for which large annotated data sets (D_s) are available. For the target domain, only a few or no instances at all have been annotated, thus the data set D_t is comparably small. Formally, we can distinguish two distributions p_s and p_t, drawn from the respective data sets. Further, $\vec{x}_i = \langle x_{i,0}, x_{i,1}, \ldots x_{i,F} \rangle \in \mathcal{X}$ will be the feature vector of instance $i \in D_{s/t}$. We assume the feature values to be mapped to real values $\mathcal{X} = \mathbb{R}^F$, where F represents the number of features. A function $o : \mathcal{X} \to \{D_s, D_t\}$ maps an instance to its origin data set. $y \in \mathcal{Y}$ represents the class label. $p_t(\vec{x}, y)$ is the distribution we are interested in.

The General Distribution In addition to the two distributions from source and target domain, Daumé III (2007) introduces a third distribution, representing the "general" domain. The union of the two data sets $D_g = D_s \cup D_t$ is drawn from this distribution p_g. The intuition behind this general distribution is that not all predictions of a (linguistic) classifier are domain-dependent: The token "the", for instance, would be tagged as a determiner in most domains.

In Daumé III (2007), a single model is trained, but on an augmented feature space. The augmented feature vector is created using $\Phi : \mathbb{R}^F \to \mathbb{R}^{3F}$ as shown in (3.1). This way, the feature vector of each instance has a general and a domain-specific part. The classifier then can learn whether to use the general domain feature set (for which it has massive training data) or the domain-specific feature set (with small training data).

$$\Phi(\vec{x}_i) = \begin{cases} \langle \vec{x}_i, \vec{x}_i, \vec{0} \rangle & \text{if } o(i) = D_s \\ \langle \vec{x}_i, \vec{0}, \vec{x}_i \rangle & \text{if } o(i) = D_t \end{cases} \tag{3.1}$$

Formally, a training algorithm then learns a linear hypothesis $\breve{h} \in \mathbb{R}^{3F}$ that contains a common, source and target specific component: $\breve{h} = \langle g_c, g_s, g_t \rangle$. In the un-augmented feature space, this corresponds to learning to hypotheses $h_s = (g_c + g_s)$ and $h_t = (g_c + g_t)$. The application of $\breve{h}$ to the augmented target sample $\Phi(\vec{x})$ is then equivalent to applying $(g_c + g_t)$ to the un-augmented sample $\vec{x}$ (Kumar et al., 2010).

Daumé III (2007) reports significant reductions in error rate for part of speech tagging, named entity resolution and chunking, compared to non-augmented ways of combining the data sets.

Harvesting Unlabeled Data An extension (Kumar et al., 2010) to the feature space augmentation approach makes use of additional, unlabeled data U_t from the target domain. In regular space, the source h_s and target h_t hypotheses are required to agree on the unlabeled data. This requirement ($h_s\vec{x} \approx h_t\vec{x}$) can be transformed into the following augmentation operation:

$$\Phi(\vec{x}_i) = \langle \vec{0}, \vec{x}_i, -\vec{x}_i \rangle \tag{3.2}$$

As these instances are unlabeled they are added once for each class label $y \in \mathcal{Y}$. Kumar et al. (2010) report results on sentiment classification using the data sets provided by Blitzer et al. (2007). Compared to the original feature space augmentation approach, they achieve a reduction in error rate between 4.3 and 39.3%.

Structural Correspondence Learning (SCL) Blitzer et al. (2006) propose a technique called structural correspondence learning. In this setting, the assumption is that unlabeled data from source and target domain are available, while labeled data is only available for the source domain. Central to SCL is the concept of *pivot features*. Pivot features behave similarly in source and target domain and occur frequently (enough). They capture the commonalities of the two domains. The technique introduces a mapping $\Phi : \mathbb{R}^F \to \mathbb{R}^{F+h}$ into a feature space that also contains h pivot features.

Using the pivot features, a number of binary pivot predictors are trained on the (unlabeled) source and target data. A pivot predictor predicts for an instance if the pivot feature is present in this instance or not. The weight vectors $\vec{w}_l$ (from training the pivot predictors) are then joined into a matrix W. After doing singular value decomposition ($W = UDV^T$), $U^T_{[1:h,:]} = \theta$ contains the top left singular vectors of W. θ is then seen as a parameter that encodes the mapping to the shared feature space. The training data is then mapped using θ into this shared feature space and appended to the original feature vector. Finally, the classifier is trained on the labeled and enhanced data. Test data can also be mapped into the feature space using Φ as defined in 3.3.

$$\Phi(\vec{x}_i) = \langle \vec{x}_i, \theta\vec{x}_i \rangle \tag{3.3}$$

Structural correspondence learning has been applied to a number of tasks: part of speech tagging (Blitzer et al., 2006), dependency parsing (Shimizu and Nakagawa, 2007), sentiment classification (Blitzer et al., 2007), parse disambiguation (Plank, 2009) e-mail summarization (Sandu et al., 2010) and dialog utterance classification (Margolis et al., 2010).

Instance Weighting J. Jiang and Zhai (2007) analyze the problem of domain adaptation by identifying two independent factors that need to be adapted. The distribution we are interested in, $p_t(\vec{x}, y)$, can be factored into $p(\vec{x}, y) = p(y|\vec{x})p(\vec{x})$. Differences between p_s and p_t can be caused by both factors: $p_t(y|\vec{x})$ may be different from $p_s(y|\vec{x})$ and/or $p_t(\vec{x})$ may be different from $p_s(\vec{x})$.

Consequently, J. Jiang and Zhai propose an adaptation that addresses both factors individually. In order to do *labeling adaptation* (adapting $p_s(y|\vec{x})$), a model is trained on target domain data D_t and then applied to the source domain data D_s. Then, the top k wrongly classified instances are removed from the source data set, as p_t apparently differs from p_s in these cases. A classifier is trained on the remaining data set D'_s. For *instance adaptation* (adapting $p_s(\vec{x})$), a bootstrapping method has been used. A model is trained on the source domain data D_s and applied to the target domain. The top k confidently predicted instances are then added to the training set and the process is reiterated. Instances from the target data set can be weighted higher. Obviously, both methods can be combined.

The results reported by J. Jiang and Zhai (2007) support the initial idea partially: Accuracy on three tasks (part of speech tagging, entity type classification and spam filtering) improves in many cases when doing labeling adaptation. However, in entity type classification, the accuracy drops when source instances are removed. Adding confidently classified target instances (instance adaptation) improves the results.

Instance preselection The system by Sagae and Tsujii (2007) achieved the highest score in the domain adaptation track of the CoNLL2007 shared task on dependency parsing. They train two different models (a maximum entropy and a support vector machine) on source domain training data D_s. Then, both models are used to parse the entire in-domain data set D_t. Sentences for which both models produce identical parses are assumed to be parsed correctly and collected in data set D_c. The maximum entropy model is then retrained on the training set $D_s \cup D_c$ and used to parse the entire in-domain data set D_t. Using this procedure, Sagae and Tsujii achieve a labeled

	Approach & Reference	Requirements
Feat. Space	Augmentation (Daumé III, 2007)	
	Augmentation++ (Kumar et al., 2010)	unlabeled data from target domain
	SCL (Blitzer et al., 2006)	unlabeled data from both domains, no need for labeled target domain data
Data Set	Instance weighting (J. Jiang and Zhai, 2007)	
	Instance preselection (Sagae and Tsujii, 2007)	Two independent classifiers
	Reliability detection (Kawahara and Uchimoto, 2008)	Reliability classifier
	Active learning (Chan and Ng, 2007)	Annotators

Table 3.1: Approaches for statistical domain adaptation

attachment score of 81.06 (unlabeled: 83.42; next best system: 80.4 LAS; both parsers individually achieve below 79 LAS on the development set).

Reliability detection Kawahara and Uchimoto (2008) improve on that by adding a component that selects reliable dependency parses. First, they split the source domain data set D_s into two parts: A training set for the parser and a training set for a reliability detector. A parser is trained on the parsing training set and an SVM model to detect reliable parses is trained on the second training set. The reliability detector uses features that indicate parse difficulty, like sentence length or number of commas. The target domain data set D_t is parsed and the SVM used to detect reliable parses. Kawahara and Uchimoto report precision 73.7% and recall 38.9% for the detection of reliable parses. Reliable parses for k sentences are then added to the source domain data D_s, the parser is retrained and D_t labeled. Using a first source domain data set, they experimentally optimize k to be 18,000, which is slightly more than the size of D_s. This way, they achieve an accuracy of 84.12 (UAS), compared to an unadapted performance of 83.58.

Active learning Chan and Ng (2007) discuss experiments employing active learning for word sense disambiguation. Using a sense annotated corpus from different newspaper genres (DSO, Ng and Lee, 1996), they iteratively train a classifier on the source domain data and apply it to the target domain data. The prediction with the lowest confidence then gets replaced with the true class, simulating actual annotation. Annotated items are weighted higher in the training procedure. The evaluation results show that (using weighting and active learning) only 4% of the target domain examples need to be annotated in order to achieve the same result as the most frequent sense baseline (61.1% accuracy).

Summary In sum, there are two groups of statistical domain adaptation approaches. The first group employs various techniques to capture commonalities in the two domains. This is done by modifying the feature space (e.g., augmenting it or adding pivot features). In the other group, the focus is on optimizing the data set and training with a regular feature space. This data set modification can be done by weighting, preselecting or removing instances.

In order to modify the feature space used by a certain NLP tool, one often needs access to the source code of the tool. The effort to integrate these feature space modifications depends on the software quality of the NLP tools involved. The manipulation of data sets is more robust in the sense that this technique can be employed without access to internals of tools.

Table 3.1 shows an overview of the approaches discussed. All approaches make the basic assumption that there is a large data set from the source domain and a small or nonexistent one from the target domain. The third column shows additional assumptions made with respect to the data sets.

Approaches on domain adaptation are hard to compare in terms of results, because there are no standard data sets. In addition, it is questionable how well an adaptation strategy that achieves improvements on one set of domains transfers to other domains. On a more fundamental level, boundaries between domains or between the notion of domain and genre seem to be vague. Nonetheless, it is clear that NLP systems need to be adapted when used on domains featuring uncommon language characteristics.

3.1.2 Word Sense Disambiguation

Because word sense disambiguation is often done in an unsupervised manner, we will discuss domain adaptation of knowledge-based unsupervised approaches to word sense disambiguation in the following. Most approaches discussed below use UKB as a base application and either adapt (i) its knowledge base or (ii) the algorithm itself. We will give a brief introduction into UKB and then discuss (i) and (ii).

UKB UKB (Agirre and Soroa, 2009) uses the PageRank algorithm in order to determine weights for candidate synsets of a given sentence. Applying the PageRank algorithm directly to the entire WordNet graph would produce a context-independent ranking of all synsets. This is due to the initialization of the vector v, which represents the probability of the i-th vertex to be hit by a random walk. In traditional PageRank, every vertex gets the same probability. In order to let the context influence the disambiguation, Agirre and Soroa add the context words as vertices to the graph and distribute the probability mass only to the context words. This way, the context words receive high initial weight.

Adapting the knowledge base The sense inventory can be adapted in a number of ways.

Adding new concepts Navigli and Velardi (2002) propose to add new senses that represent domain-relevant multi word expressions. Initially, multi word candidates are extracted from a domain corpus and filtered using the information-theory based measures domain relevance and domain consensus (Velardi et al., 2001). The assumption is then that a word x subsumes the multi word wx, i.e., that longer multi word expressions are more special than shorter ones. Therefore, a new sense wx is added to the hierarchy as a hyponym of the sense representing x. The system was put to use in order to speed up the process of ontology creation for the tourism domain. Navigli and Velardi report a precision of 85% for the semantic disambiguation of multi word expressions.

Reranking concepts WordNet senses are ranked according to their frequency in a corpus. Navigli (2009, p. 10:45) reports an accuracy of 57% on a mixed-genre corpus (Senseval-1) for a word sense disambiguation system that always assigns the most frequent sense. It is reasonable to assume that the ranking of senses is highly domain-dependent. Therefore, McCarthy et al. (2004) employ ways of computing the most frequent sense from a new domain corpus ("predominant sense") and rerank the senses accordingly. First, a thesaurus is created from an automatically parsed domain corpus (Lin, 1998). From this thesaurus, the k nearest neighbors for each target word w and distributional similarity scores between w and its neighbors are extracted. Let $N_w = \{n_1, n_2, \dots, n_k\}$ be the list of neighbors, $\{\mathrm{dss}(w, n_1), \mathrm{dss}(w, n_2), \dots, \mathrm{dss}(w, n_k)\}$ be the set of distributional similarity scores and $\mathrm{senses}(w)$ be the set of senses of word w. The *prevalence score* ps for a specific word sense $s_{w,i}$ is then calculated as shown in Equations 3.4 and 3.5.

$$\mathrm{ps}(s_{w,i}) = \sum_{n_j \in N_w} \mathrm{dss}(w, n_j) \frac{\mathrm{wnss}(s_{w,i}, n_j)}{\sum_{s'_w \in \mathrm{senses}(w)} \mathrm{wnss}(s'_w, n_j)} \tag{3.4}$$

$$\mathrm{wnss}(s_{w,i}, n_j) = \max_{s'_{n_j} \in \mathrm{senses}(n_j)} \begin{cases} \mathrm{lesk}(s_i, s'_{n_j}) \\ \text{or} \\ \mathrm{jcn}(s_i, s'_{n_j}) \end{cases} \tag{3.5}$$

McCarthy et al. (2004) experiment with using lesk (Banerjee and Pedersen, 2002) and jcn (J. J. Jiang and Conrath, 1997) as similarity measures between WordNet senses (3.5).

Reddy et al. (2010) conduct a series of experiments in which prevalence scores extracted on domain-specific corpora are used to initialize the link weight between context words and candidate synsets. This leads to an improvement of about 10% precision and recall (compared to using default link weight).

Removing and aggregating concepts There are several approaches that focus on specific parts of WordNet (Core WordNet) or merge multiple existing senses into one (OntoNotes). Core WordNet (Boyd-Graber et al., 2006) contains the most salient and basic synsets for the most frequent lemmas in the BNC. The creation of the OntoNotes (Hovy et al., 2006) resource is guided by the inter-annotator agreement: As long as the inter-annotator agreement is less than 90%, the sense definitions are revised (i.e., senses are merged). Having fewer choices for a given lexeme makes the task easier. To our knowledge, removing and aggregating senses has to not been used for the purpose of domain adaptation, but merging or removing domain-irrelevant senses would be a form of domain adaptation.

Adapting the word sense disambiguation algorithm

Initialization Reddy et al. (2010) propose several alternatives to the initialization of the weight vector v in the page rank algorithm and use UKB for their experiments. First, they introduce the keyword ranking score krs that represents the "keyness" of a word for a specific domain. The keyword ranking score is calculated as shown in (3.6), where LL represents the log-likelihood ratio as described in Rayson and Garside (2000).

$$\mathrm{krs}(w) = \frac{LL(w)}{\sum_{w_i \in \mathrm{words}(d)} LL(w_i)} \tag{3.6}$$

The context words are then initialized with krs instead of uniformly. Reddy et al. report a minor improvement in precision and recall (+ ~ 1%).

Context choice Stevenson et al. (2012) propose to change the set of context words that is used as input to UKB. The approach assumes that the domain of the target text is known and that a domain corpus is available (in this case, a domain is described by the so-called medical subject heading, MeSH). Several methods are used to extract key terms for the domain from the domain corpus. These key terms are then added as contexts for UKB. Overall, they achieve an improvement in accuracy of 3.3 percentage points when using relevance feedback (Rocchio, 1971) and inverse document frequency for the extraction of key terms.

Summary Again, comparing different approaches on domain adaptation for word sense disambiguation is difficult, even when they are all using the same word sense disambiguation system, because the data sets and underlying assumptions are different. Deciding on a specific approach on word sense disambiguation adaptation should take into account what the actual aim is and what resources are available. Calculating prevalence scores, for instance, requires a large corpus, which is not always available. If structured domain knowledge is available in some form, the manipulation of the knowledge base to incorporate this domain knowledge may be feasible. In any case, UKB makes these kinds of manipulation straightforward to implement.

3.2 Computational Narrative Analysis

In this section, we describe the related work in the area of computational narrative analysis. We will first give some background information (Section 3.2.1) on narratives and narratology. The remainder of the section is split into three parts: Section 3.2.2 describes work which focuses on modeling individual narratives in a deep, fine-grained way. As we are ultimately interested in comparing and aggregating narratives, we will discuss approaches towards this aim in Section 3.2.3. The story intention graph framework includes a modeling as well as an aggregation part and is therefore described in two parts. We will give an overview of the discussed approaches in a schematic form at the end, in Section 3.2.4.

3.2.1 Narratological Background

Story telling and narratives have been researched in the discipline of narratology. Mani (2012) describes narratology as the theory of narrative structure and narrative structure as "representations of different phenomena that are relevant to making sense of narrative as story" (Mani, 2012, p. 4). In order to understand a narrative, humans have a certain understanding of several aspects of the narrative. In order to "computationally understand" a narrative, these aspects need to be represented. Mani mentions five aspects in particular: The narrator, narrative levels (embedded narratives), audience, time and fabula. This work focuses on the aspect of fabula. A narrative fabula is a "chain of events (actions, happenings), along with existents (characters, items of setting)" (Chatman, 1980). The notion of a chain implies connections of some sort between the events.

Forster (1927) distinguishes between *story* ("a narrative of events arranged in their time sequence") and *plot*: "Also a narrative of events, the emphasis falling on causality" (both are fabulas, cit. Mani (2012)). This notion of causality is worth explaining, because it employs a rather loose sense of causality.

(1) a. The king died and then the queen died.

b. The king died and then the queen died of grief.

A simple "list" of events, as in Example 1a would be a story, but not a plot according to Forster. If, however, the events are connected so that a causal connection between the events is expressed, they form a plot. Example 1b shows a plot, because the second event is causally related to the first. In this thesis, we will use the term *story* in Forster's sense.

3.2.2 Story and Plot Models

Story Grammars

The story grammar approach, as implemented by Correira (1980), represents events in a story as propositions. Initially, temporal relations are added to represent the temporal ordering of events. By employing a collection of rules, a set of propositions is then connected to a more abstract representation of the events. E.g., the events described in Example 2 are connected to the meta proposition in Example 3 ("x makes a trading voyage to y").

(2) buy(x,ship), buy(x,goods), load(x,ship,goods), sail(x,to:y,means:ship)

(3) tradingvoyage(x,with:goods,in:ship,to:y)

Such a rule does not only contain a concrete list of events and an abstract description, but may also contain pre- and postconditions.

The requirement of a rule base limits the applicability of this approach. Manual creation of knowledge bases is labor-intensive and expensive and rules as fine-grained as in 3 can be expected to be heavily domain-dependent. Although existing knowledge bases such as SUMO (Niles and Pease, 2001), Cyc (Lenat, 1995) or FrameNet (Fillmore et al., 2003) contain such script-like knowledge to a certain extent, their coverage is severely limited. In particular with respect to domains from the humanities, many rules would need to be written. First attempts on semi-automatic acquiring of scripts have been made (e.g., Regneri et al. (2010)) that could presumably be extended towards specific domains.

Plot Units

Plot units are described and introduced by Lehnert (1981). A plot unit consists of affect states and links between affect states. An affect state is always bound to a specific participant and may be an event with a positive (+) or negative (-) effect or a mental state ($\mathcal{M}$) without effect. Lehnert distinguishes four different link types that are used to connect affect states:

Motivation (m) A causal relation between mental states

Actualization (a) A mental state gets realized and has positive or negative effect

Termination (t) The affective impact of an event ends

Equivalence (e) If multiple perspectives are separated, this relation represents that the same event has both positive and negative effects

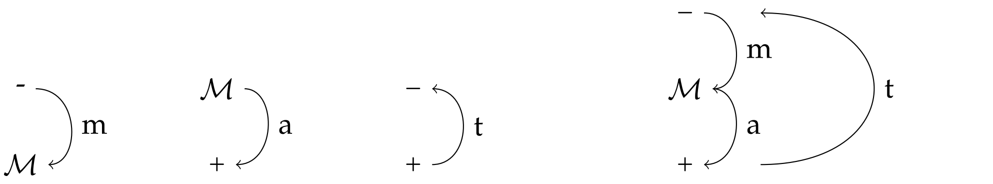
There are 15 different ways of linking two states with a link, because not all links are compatible with all node types. These 15 pairwise configurations are called primitive plot units and represent typical situations like *resolution*, *success*, etc. More complex plots can be put together by combining primitive plot units.

(4) a. Mary got fired and needs a job.

b. She successfully applies for a job.

Both sentences in Example 4 describe two states: 4a describes a state with a negative effect (the firing event) and a mental state (Mary needing a job). In 4b, the need for a job (implied by the application) is a mental state. The fact that she successfully applies for a job introduces a state with a positive effect.

The story thus contains the three primitive plot units *problem* (in 4a), *success* (in 4b) and *resolution* (in both, surpassing the mental state and directly linking the firing to the



(a) Problem (b) Success (c) Resolution (d) Complex Plot Unit: Problem Resolution

Figure 3.1: Primitive and complex units (Lehnert, 1981)

hiring event). Figure 3.1a, b and c show them in a graphical form. Figure 3.1d shows how they are combined into the complex plot unit *intentional problem resolution*.

Most narratives involve multiple characters. Therefore, plot units can include cross-character links, i.e., links between states of different characters. Lehnert also describes a number of typical configurations involving cross-character links, like *request* or *threat*.

Automatic plot unit recognition A system that automatically detects plot units in narratives has been proposed by Goyal et al. (2010). To our knowledge, this is the only system for automatic plot unit recognition.

The algorithm works in four steps: (i) A dictionary is used to identify verbs that represent an affect state. Goyal et al. (2010) experiment with various dictionary sources, including FrameNet (Fillmore et al., 2003) and sentiment-based resources. (ii) The characters are identified and their coreferences resolved by use of a simple, rule-based coreference resolution system. The system assumes that each story only contains two different characters and that both characters are mentioned in the title of the story. (iii) For the mapping of affect states to characters, the Sundance parser (Riloff and Phillips, 2004) is used to obtain a shallow syntactic parse of each sentence. Then a number of rules is used to determine the characters for which the affect state holds. (iv) If two characters have affect states induced by the same word, a (cross-character) link between the events is created. Links for a single character are created between pairs of consecutive affect states.

For evaluation, a gold standard was created that consists of 34 fables of Aesop, annotated by two authors and adjudicated by the third. Inter-annotator agreement is not reported for the annotation of links. Goyal et al. (2010) report results for a number of configurations in terms of precision, recall and f-score. The best performance they achieve for the detection of affect states is an f-score of 45. The heuristics for identifying links achieves between 72 and 92 f-score on gold affect states, depending on the link type. On system affect states, the performance is between 5 and 25 f-score.

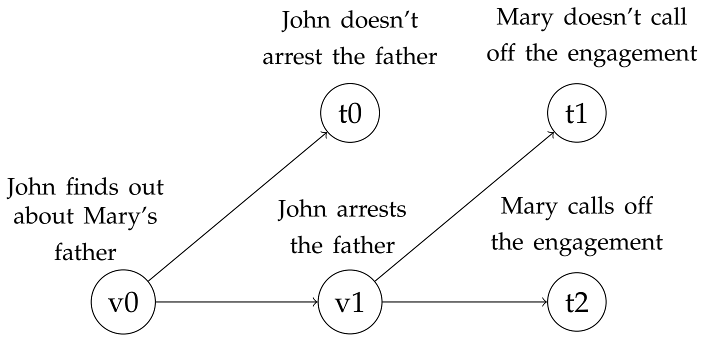

John doesn't arrest the father
Mary doesn't call off the engagement
t0
t1
John finds out about Mary's father
John arrests the father
Mary calls off the engagement
v0
v1
t2

Figure 3.2: Game tree for the engagement story (Mani, 2012)

Doxastic Preference Framework

Having roots in game theory, the Doxastic Preference Framework (DPF) (Löwe and Pacuit, 2008) can be used to model the beliefs of characters in a story. Central to the DPF is a tree, in which each node represents a decision point for a character (game tree). Based on his or her beliefs about the preferences of him- or herself and other characters, he or she makes a decision. Depending on the outcome of the decision, other characters are forced to change their beliefs and make decisions on their own.

(5) John was thrilled when Mary accepted his engagement ring. But when he found out about her father's illegal mail-order business, he felt torn between his love for Mary and his responsibility as a police officer. When John finally arrested her father, Mary called off their engagement.

Figure 3.2 shows a game tree for the engagement story (5, example and figure taken from Mani (2012)). The first decision point is for John, when he found out about the criminal activities of Mary's father. Either he arrests the father (node v1) or he does not (node t0). Depending on the outcome of John's decision, Mary has to decide whether she calls off the engagement or not (node t1 vs. t2). The DPF defines various event types, such as *expected event*, *unexpected event*, *Betrayal* etc. In addition to the "decision structure" of the story, the tree is used to formally represent beliefs and preferences for each character, e.g., in the form of a function ordering possible situations by preference for a character (see Löwe and Pacuit (2008) for the formalization).

To our knowledge, a single annotation of a narrative has been performed. Andel (2010) formalized seven different episode fragments of a TV crime series, but gives neither annotation guidelines nor inter-annotator agreement. He does, however, mention that the framework does not offer any means to represent ambiguities or underspecifications in the story. In his case, one episode offers two possible perpetrators for a kidnapping. The formalization forces to decide on one of the two, as the following parts of the story need to be encoded differently depending on who did the kidnapping.

No experiments on automatic extraction of game trees have been published.

Story Intention Graphs

Story intention graphs (Elson, 2012b) are graph-based representations of textual narratives that focus on the intentions of characters. The graphs are multi-layered and model the text itself as well as its story timeline and intentions, plans and goals of the participants.

The *textual layer* contains the original text, broken down into fragments (clauses or sentences). The *timeline layer* contains an abstract representation of events and statives appearing in the story as proposition nodes. Proposition nodes not only model events happening in the story reality, but also modal propositions like uncertain, imagined or believed concepts. Each proposition node is related to state nodes which represent points in (story) time by use of the temporal relations BEGINS AT and ENDS AT. Proposition nodes are related to the text nodes they interpret with the relation INTERPRETED AS (IA).

The *interpretative layer* serves as a place for interpretations of the story: It contains the understanding the reader or listener gets while comprehending the story. Thus, it does not only include content stated explicitly in the story, but also content inferred by the annotator, reader or listener. The nodes in the interpretative layer can be of the types *Belief*, *Goal*, *Interpretative proposition* or *Affect*. Nodes can also be distinguished according to the actualization statuses they are in at points in story time: Some nodes are true with respect to the story world and time, others are false and some are hypothetical (their actualization status has not been determined). The interpretative layer is connected with the timeline layer by adding relations between nodes in the layers. A relation can be of one of 13 different types, some of which have impact on the actualization status of the nodes they connect.

Encoding For three collections of stories, intention graph encodings have been collected. Collections A and B contain a selection of stories by Aesop. The criterion for the selection was a clear timeline and story events that are causally connected. Collection A contains two encodings for each of 20 fables, but covers only the textual and timeline layers. Collection B contains 6 additional fables and also the interpretative layer for all 26 fables. More than two encodings for most of the stories are available in B. Collection C contains encodings of the timeline and interpretative layer for eight stories of very different lengths and genres, indicating that the formalism is applicable to not only fables. Table 3.2 shows an overview of the different collections with several key properties.

Elson (2012b) gives a detailed report on the manual labor invested, which is an important aspect for any kind of annotation project. For the collection A (without interpretative layer), the time to encode a fable dropped from several hours to 30-45 minutes due to the training effect. The median time spent on encoding a fable in collection B (with interpretative layer) or a story in collection C was one respectively two hours. In addition, 2-3 hours of training were invested for each collection.

For collection A, which only contains the textual and timeline layers, Elson (2012b) reports that 10% of the proposition pairs (from two annotators) are fully identical (El-

	A	B	C
Number of texts	20	26	8
Annotators per text	2	1-3	1
Text sources	Aesop		Various
Text length	⌀125 words		$1,149 - 25,649$

Table 3.2: Overview of collections with annotated stories

son, 2012b, p. 201). For measuring agreement in the collections B and C (that include the interpretative layer), a set of 80 patterns encoding typical situations is used (Elson, 2012a). A vector is constructed for each graph by setting a value of 1 if a pattern occurs in the encoding and 0 otherwise. This way, agreement is quantified by measuring the cosine similarity for different encodings. Encodings of the same source texts made by different encoders have a significantly higher cosine similarity between their feature vectors than encodings of different source texts. In terms of Cohen (1960)'s kappa, the agreement is $\kappa = 0.55$.

We will discuss the use of story intention graphs for comparison of narratives in the next section.

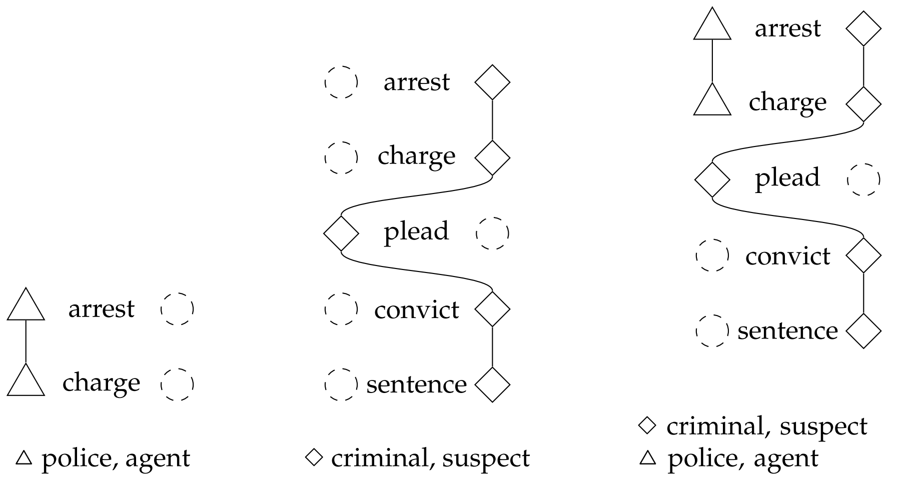
Narrative Schemas

Narrative schemas have been proposed by Chambers (2011) as a script-like structure that can be extracted automatically from texts. A narrative schema describes situations consisting of multiple events and participants, similarly to complex plot units or meta propositions.

A narrative schema consists of multiple narrative chains. A narrative chain represents a (partially ordered) set of events involving a single protagonist in specific (grammatical) roles. Such a chain is called typed, if the protagonist is of a certain (semantic) type. The formalism does not restrict itself to a specific type system or hierarchy, but uses lexemes extracted from the texts in order to represent the type.

Figure 3.3 shows two typed narrative chains (a and b). The chain in 3.3a shows an entity of the type police or agent participating in arrest- and charge-events, and in both cases as subjects. Similarly, in 3.3b, we see a number of events in which an entity of the type criminal or suspect participate – in plead-events as subject, in all others as objects. A narrative schema that is constructed by merging a and b is shown in 3.3c.

Chambers (2011) describes an unsupervised algorithm to extract chains and schemas from texts. The algorithm relies on coreference chains and dependency parses to detect possible event chains. For each coreference chain (i.e., each entity), a list of pairs (v, d) is extracted, one for each mention of the chain. v represents the verb of which the mention is an argument and d the syntactic dependency of the mention. Then, the point-wise mutual information (pmi) between two event/role pairs can be approximated by counting the number of times two verbs share a coreferring entity in specific syntactic roles in a large corpus.



(a) Narrative chain (b) Narrative chain (c) Narrative schema

Figure 3.3: Narrative chains and schema (Chambers, 2011)

The most probable next event in a chain can be predicted by maximizing the pmi for verbs in a specific document. In order to extract a (semantic) type for an entity, the most salient (i.e., most often used) head words of the coreferring mentions are used to represent the type of the entity. Prediction of the next most probable event can then be extended to include similarity of the type.

As an evaluation, Chambers introduces the narrative cloze task: A single event is removed from a known event chain (or schema). Ideally, the missing event is among predicted events. In order to compare the outcome, Chambers reports the rank of the missing events within the list. The NYT portion of the Gigaword corpus has been used for the experiments. When using typed narrative schemas, The average ranked position of the removed event is at approximately 72% of the result list: If the system proposes a (ranked) list of 100 events, the correct one will be at the 72th position. This is an improvement over using untyped schemas or chains.

3.2.3 Comparison and Aggregation

We will now focus on approaches for the automatic comparison and aggregation of narratives. We will discuss the non-technical work by Propp (1958) in Section 4.1.

Sequence Alignment

An important aspect of comparing and aggregating multiple narratives is the identification of similar events in similar contexts. This is highly related to the notion of sequence alignment, which has been researched intensively in the area of bio informatics for aligning protein sequences. A classic algorithm for pairwise sequence alignment is the Needleman-Wunsch algorithm (Needleman and Wunsch, 1970) . We will discuss it in detail in Section 6.2.1. The algorithm generates a global alignment (i.e., every element in both sequences is processed and evaluated) and an alignment score. The global alignment does not include crossing links, but may contain gaps and mismatches.

Applications for Alignment of Narratives One very recent use of the Needleman-Wunsch algorithm in the context of story comparison has been published by Fay (2012). One of the issues of the Needleman-Wunsch algorithm is that sequence elements are atomic to the algorithm. Any internal structure that the sequence elements may have are ignored by the algorithm, if it is not captured with a similarity function. In addition, each (possible) link is processed in isolation. Its similarity score only depends on the two elements to be compared and is ignorant about other linked sequence elements.

Fay's sequence elements are predicate argument structures. The goal of the algorithm is not only to link the events of two stories, but also to link the participants (persons or objects) of the stories to their counterparts. To this end, Fay uses two algorithms simultaneously: Sequence alignment according to Needleman-Wunsch and the construction of a match tree. The latter makes sure that role fillers of the events to be linked are also matched in a consistent manner. A node in the match tree consists of two sets of unbound role fillers from both stories and a list of bindings with pairs of role fillers. A binding represents role fillers that are considered to be corresponding. At the beginning, the binding list of the root node is empty. In each step, nodes are added for possible pairings of objects until both lists of unbound objects are empty.

The similarity function that is employed by the Needleman-Wunsch algorithm takes the binding list into account and can reject alignments that contradict the existing binding list. The important idea is that not all possible pairings of objects are added, but only the one for which the sequence alignment algorithm produces the highest score. The similarity for two predicate argument structures is 0, if one of the objects used in the same role in both events is already bound. Otherwise, a similarity score based on WordNet is used.

Obviously, this technique greatly reduces the computational complexity of finding the optimal binding list, compared to brute force methods. An evaluation of the qualitative performance of the technique has not been reported, the evaluation focuses on technical aspects like reduced running time.

Detection of Analogous Story Intention Graphs

Elson (2012b) describes three different algorithms to detect similarities and analogies based on story intention graphs (see previous section). The first one is a relatively simple alignment algorithm that works on the timeline layer. The second one calculates the overlap of previously defined situation patterns. The third algorithm is based on the analogical constraint mapping engine (Holyoak and Thagard, 1989) and tries to find correspondences in the story intention graphs without predefined patterns. All algorithms assume that at least partial SIG encodings for the stories have been created.

Propositional and temporal overlap The first algorithm serves as a baseline algorithm and only uses the timeline layer. It works on pairs of (story intention) graphs. The algorithm iteratively links the two propositions that are most similar, as long as no constraints are violated. The set of constraints includes (i) a similarity threshold, (ii) disallowing crossing links and (iii) consistent role fillers, which are updated every time a new alignment link is created. The algorithm terminates if no proposition pairs are left that fulfill the constraints.

The similarity function used by Elson is based on three different features: WordNet similarity, morphological derivations and synonymy/antonymy. (i) WordNet similarity is measured using the Lin (1998) definition of WordNet similarity. It sums the similarity scores of the two predicates and the arguments. (ii) For measuring morphological derivations, a heuristic is implemented that matches a stative proposition with a modifier attached to an event proposition. An appropriate similarity value is given, if the propositions are matching and thus considered paraphrases. (iii) Lastly, VerbOcean is used to detect (indirect) synonyms and antonyms.

As a gold standard for the evaluation, Elson (2012b) collected ratings for 2,700 proposition pairs out of collection A (see above) using Amazon's Mechanical Turk. The annotators were given a pair of natural language sentences and asked to rate if the sentences are paraphrases of each other. The pairs were pre-filtered to not include pairs with a large difference in relative position in the story (more than 40%). Although the alignment algorithm performed better than the Jaccard index baseline, the improvement shown is not significant.

Static pattern matching In the second algorithm, previously defined situation patterns are used. A situation pattern is a hypothetical story intention graph "which minimally describes a certain narrative scenario" (Elson, 2012b, p. 218) and is the intention graph equivalent of a plot unit.

A feature vector created for each story shows for each of the 80 different patterns if it is present in the story or not. The general idea is then that if two story encodings share a certain amount of patterns, they are similar and have analogue parts. Comparing the cosine of the feature vectors of story encodings for the same story (created by different encoders) with the cosine of the feature vectors of encodings for different stories, it can be shown that the cosine similarity for encodings of the same story is significantly higher than for encodings of different stories.

Dynamic analogy detection The third algorithm is based on the Analogical Constraint Mapping Engine (ACME) proposed by Holyoak and Thagard (1989). It does not rely on predefined patterns but instead detects overlap in the story encodings directly. Before starting the algorithm, (transitive) closure rules are applied to both graphs.

The algorithm uses "globs" as its core data structure. A glob represents a potential alignment and contains a binding list consisting of nodes and agents. Initially, every pair of (proposition) nodes on the timeline layer of the two input graphs is put into a glob. Then, each glob is expanded by adding interpretative unseen nodes, if they

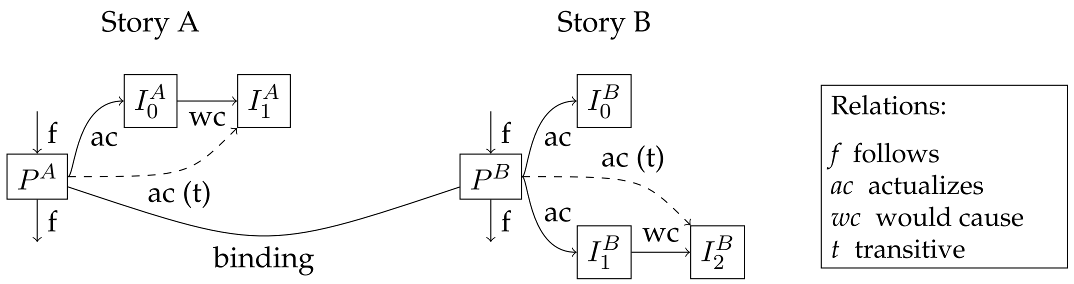


Figure 3.4: An example glob for dynamic analogy detection (Elson, 2012b)

can be reached via the same relation following the arc directions. If there are multiple outgoing relations, a glob forks and each possibility is considered separately. Each time a glob is expanded, its binding list is updated.

Figure 3.4 shows a single glob initialization for the stories A and B, containing proposition nodes P, interpretative nodes I and several ACTUALIZES (AC) and WOULD CAUSE (WC) relations. Two relations are added by applying the closure rules by transitivity (t).

The proposition nodes P^A and P^B are put in a glob. Initially, only P^A and P^B are on the binding list. After the first expansion following the AC relations, the glob has been forked in six globs, each containing one of the following, additional bindings: (I_0^A, I_0^B), (I_0^A, I_1^B), (I_0^A, I_2^B), (I_1^A, I_0^B), (I_1^A, I_1^B) and (I_1^A, I_2^B). The glob containing the binding (I_0^A, I_1^B) can be further expanded by following the relation WC. After that, this largest glob contains the bindings $\{(P^A, P^B), (I_0^A, I_1^B), (I_1^A, I_2^B)\}$. In total, we have expanded the initial single glob into six globs, one containing three bindings, the other containing two bindings.

If a glob can not expand any further, the pairs of proposition nodes consistent with its binding list are determined by applying the Needleman-Wunsch algorithm on the proposition nodes. This way, the alignment with the highest number of compatible links can be calculated for each glob. After this step has been completed, a number of possible alignments has been determined. Starting with the alignment with the maximal number of links, the alignments are now merged if they contain compatible bindings. The final result is a list of mutually incompatible alignments, sorted by their size (i.e., the alignment linking the highest number of nodes is ranked first).

Evaluation Two experiments are carried out to compare the algorithms directly. Both of them make use of Amazon Mechanical Turk (AMT) to get ratings.

In the first setup, the AMT users were asked to rate the analogy an algorithm extracted from two stories. The users were displayed a textual representation (created using rules) of the analogies and both stories. The rating was collected on a 3-point Likert scale for two questions, one about the accuracy and one about the completeness of the analogy. For approximately 100 story pairs, three ratings have been collected for the output of each algorithm. In 61% of the cases, a 2:1 majority occurred, the rating was consensual in 27% of the cases. The results show that the propositional overlap and the dynamic analogy detection algorithms achieve the highest accuracy (differences between them statistically insignificant). The best completeness rating is achieved by the static pattern algorithm, closely followed by the dynamic one.

The second setup collected bare similarity ratings for a story pair. The participants read both stories and then rated their similarity on a 3-point Likert scale[1]. Full agreement was achieved in 46.3% of the cases, 50.4% show a 2:1 majority. A linear regression model with predictor variables from the different algorithms was trained and the correlation evaluated. Here, the results indicate that propositional overlap is the weakest algorithm. The highest correlation (Pearson's $r = 0.33$) is achieved by using features from the static and dynamic pattern detection algorithms in combination, the features from the propositional overlap algorithm do not make a difference.

Elson published the annotated stories under the name DramaBank, but this does not include the similarity ratings collected for the experiments.

Predicate Alignment System

The predicate alignment system, as described in Roth and Frank (2012), has been developed to align predicate argument structures in comparable texts. The system works by generating a graph in which predicate argument structures from both documents are represented as vertices. Then, pairwise similarity between the vertices is calculated and weighted edges between their vertices are added. By applying a minimum cut algorithm, the graph is then cut in two parts, such that the summed weight of removed edges is minimal. This cutting is repeated until each subgraph contains at most two vertices.

The algorithm has been applied to newspaper texts and evaluated against a manually created gold standard featuring 70 document pairs, each document contains between 100 and 300 words. The system's results have been compared against two baselines. In the first baseline, same lemmas have been aligned. The second baseline uses a word alignment tool that has been developed for statistical machine translation (Berkeley Aligner, Liang et al. (2006)), based on automatically detected paraphrasing sentences. The system outperforms both baselines in terms of precision and f-score, while the word alignment tool baseline achieves a higher recall.

We will discuss the predicate alignment system in more detail in Chapter 6.

Bayesian Model Merging

Bayesian model merging (Stolcke and Omohundro, 1993) has been proposed as a technique for the induction of a hidden Markov model (HMM) from a set of sequences. Finlayson (2012) uses Bayesian model merging to create a merged representation for multiple, analogous stories. We will first focus on the algorithm itself and discuss its application to narratives afterwards.

[1] Users were also asked to provide a textual description of the similarity, but we disregard that here.

The Core Algorithm Let $\mathcal{S} = \{S_0, S_1, \dots, S_n\}$ be a set of input sequences of variable lengths, such that $\forall S \in \mathcal{S} : S = \langle s_0, s_1, \dots \rangle$. Given the input sequences $\mathcal{S}$, the goal of the algorithm is to maximize the probability of the model given the sequences: It searches for a maximally probable model M: $\arg\max_M P(M|\mathcal{S})$. By application of Bayes' theorem, this can be rewritten as $\arg\max_M P(M)P(\mathcal{S}|M)$. The algorithm works iteratively after an initialization.

The HMM M_0 is initialized such that $\forall S_i \in \mathcal{S} : P(S_i|M_0) = \frac{1}{n}$. In words, all sequences are equally probable. Then, the algorithm merges two hidden states of model M_i in order to induce model M_{i+1}. The two states are selected such that $P(M_{i+1}|S) > P(M_i|S)$. Each merge introduces new transitions into the HMM. Therefore the number of paths through the HMM increases, which in turn decreases the probability for the sequences: $P(\mathcal{S}|M_i)$ monotonically decreases. Therefore, the prior $P(M_i)$ needs to be defined in such a way to compensate for that and to control the merge operations.

Application to Narratives Finlayson (2012) uses this technique in order to automatically detect structural similarities in narratives. For this application, the observed states of a HMM represent story events and hidden states the unobserved event structure. Using a prior based on a geometric function (3.7) Finlayson applies the algorithm to a corpus of 15 fairy tales which have been manually labeled with semantic roles and coreference chains.

$$P(M) = p(1-p)^{|M|-1} \prod_{\forall n \in M} K(n) \tag{3.7}$$

$$K(N) = \begin{cases} 1 & \text{if } \mathrm{sim}(N) == \mathit{true} \\ t & \text{otherwise} \end{cases} \tag{3.8}$$

Finlayson used two similarity functions in succession, i.e., the algorithm is used twice with different functions (plugged in the same prior function shown in (3.7)). In both cases, the similarity function is defined to measure the similarity of all events emitted from a single state and similarity is measured as a boolean value: The events fulfill the similarity criterion or not. The similarity functions work on automatically assigned, but manually corrected annotations on many levels, including Propbank frames as event representations, semantic roles, word senses and coreference annotation. The annotation also includes the assignment of discourse entities to character functions according to Propp (1958)[2].

The first stage focuses on semantics and uses four different similarity criteria. If all criteria must be fulfilled, the events are considered similar. (i) All events must be "non-generic". Finlayson defines an event to be generic according to the WordNet sense its target has. If the sense is a hyponym of communication, perception or motion, the event is considered to be generic[3]. This is to exclude verbs like *say* from being merged. (ii)

[2] Proppian character functions are prototypical roles of discourse entities, like hero or villain. See Section 4.1 for details.

[3] To be clear: This notion of genericity is not the same as the one discussed in Krifka et al. (1995) and we aimed at in Reiter and Frank (2010).

Name & Reference	Autom.	Characteristics
Story Grammars (Correira, 1980)	–	Grammar-like structures on events
Plot Units (Lehnert, 1981)	part.	Modeling of positive or negative effects of events for characters
Story Intention Graphs (Elson, 2012b)	–	Models beliefs and intentions of characters
Doxastic Preference Framework (Löwe and Pacuit, 2008)	–	Models preferences and expected outcomes of actions

Table 3.3: Story modeling approaches

All pairs of events must be "synonyms". A pair of events is defined to be synonymous if their assigned WordNet senses (or hypernyms of the senses) share at least one synonym. (iii) Each PropBank frame that is assigned to an event must be assigned at least twice (within the state). This condition works as a balance for the more loose synonymy requirement. (iv) All pairs of events must feature consistent use of character functions in the semantic roles of the event. This condition requires that the character functions of semantic roles should appear in compatible ways in different events.

In the second stage, the similarity function focuses on the valence of the events and uses two criteria. (i) The character function assignments must be compatible (this is the same requirement as in the first stage). (ii) The events must agree in their valence, i.e., all events have the same number of arguments.

The algorithm's performance was compared to manually produced gold standard annotations in the style of Propp's event functions (see Section 4.1). He evaluates the clustering of events into Proppian functions with the chance-adjusted Rand-index (Hubert and Arabie, 1985). The performance score ranges from 0.51 in the most strict setting to 0.71 in the most lenient setting.

As we are using Bayesian model merging in our own experiments, we will discuss it in more detail in Chapter 6, although we will be using different similarity measures.

3.2.4 Summary

Table 3.3 gives an overview of the approaches on *modeling individual stories* discussed above. The middle column indicates whether automatic generation of these models has been investigated, the last column shows a short description of what the approach models.

All approaches have in common that they are very expressive. A (large) collection of these models would undoubtedly enable interesting empirical research. However, manual annotation in these frameworks is time-consuming and expensive and automatic annotation seems to be out of reach at a reasonable quality level. Another issue that some of the fine-grained modeling approaches have is that they force annotators or encoders to decide on a single meaning, even if the story is in fact underspecified or ambiguous. This makes encoding difficult and encoder agreement hard to measure.

The automatic modeling of these structures gets even more difficult if there is room for interpretation. This raises also questions about evaluation, because a disagreement between annotators (or between a system and an annotator) could just mean that another possible interpretation has been modeled.

The approaches in Table 3.4 focus on finding commonalities across different narratives. The table contains name and reference, whether the approach has been used on automatically processed texts, prerequisites that the approach has, whether data sets have been released and the key characteristics. All of them are unsupervised approaches.

Sequence alignment is a very basic approach that has its (original) focus not on language data. We therefore give neither data sets nor automatization. The extension of the raw sequence alignment algorithm to also generate bindings of participants of events (*sequence alignment + binding list*) does rely on linguistically annotated texts, in particular semantic roles. Although a technical evaluation has been done in the form of a complexity study, no qualitative evaluation of either the alignment nor the binding list has been published. Similarly, neither the implementation nor the data set is available.

Narrative schemas are an unsupervised approach and they are extracted from fully automatically annotated texts. However, the approach relies on the existence of a large corpus (in this case: Gigaword) from which pointwise mutual information can be calculated. This makes an application of this approach in digital humanities difficult, as large corpora are often not available. The extracted event schemas are available. To our knowledge, the implementation has not been released.

The *predicate alignment* system does rely on linguistically processed texts on a number of levels including PropBank semantic roles and coreference resolution. The system has been used on automatically annotated texts and the data set has been published. The system is currently not publicly available, but we are in close contact to the author.

The two approaches that are most closely related to our work are story intention graphs (Elson, 2012b) and Bayesian model merging (Finlayson, 2012), because both have their focus on analogy detection on narrative texts.

Story intention graphs aim at modeling the intentions of story characters in graphs. The framework allows the discovery of "deep" analogies across texts, including analogies of intentions and beliefs which are not even mentioned in texts but interpreted by an encoder/annotator. An obvious prerequisite for finding analogies across intention graphs is that both narratives are encoded as intention graphs. Automatic encoding of these graphs is currently out of reach for NLP, and, given the non-linguistic aspects of these graphs, may remain out of reach for some time. The encoded story intention graphs (for the collections described above) are available together with the encoding application.

Bayesian model merging, in contrast, builds on mostly linguistic annotations of texts, but some of the similarity measures also require significant domain specific pre annotations (character functions). Finlayson based his experiments on semi-automatic annotations. Automatic linguistic processing has been done, but for the experiments on Bayesian model merging, the annotations have been manually corrected. In order to

Name & Reference	Autom.	Prerequisites	Data	Characteristics
Sequence alignment (Needleman and Wunsch, 1970)		Similarity, gap cost	–	Global, pairwise alignment
Sequence alignment + binding list (Fay, 2012)	✓	Linguistic annotations	no	Adds generation of a binding list to Needleman and Wunsch (1970)
Narrative Schemas (Chambers, 2011)	✓	large corpus	yes	Prototypical situations, extracted from a large corpus
Predicate alignment (Roth and Frank, 2012)	✓	Linguistic annotations	yes	Graph based detection of 1-to-1-links of predicate argument structures
Analogous Story Intention Graphs (Elson, 2012b)	–	SIG encodings	yes	Detection on analogies on across story intention graphs
Bayesian model merging (Finlayson, 2012)	–	Ling. annotations, character functions	no	HMM for aggregating similar sub sequences

Table 3.4: Story aggregation approaches

do large-scale empirical research on narratives, such a manual correction is infeasible. Neither the data sets nor the implementation is available.

4 Application Scenarios

In this chapter, we will discuss two different application scenarios for structural event analysis of narrative texts. Both scenarios come from the area of folklore research, i.e., research on cultural heritage. We will give a brief introduction to each research area and highlight relevant research questions. We will further discuss how computational narrative analysis techniques can be beneficial with regard to these questions. Finally, we will present corpora we have collected for both areas.

4.1 Folktales

Folktales are tales that have been passed down orally for a long time and are part of the folklore and *cultural heritage* of a culture or group. Folklore has been studied in the area of folkloristics and literary sciences.

Fairy tales, fables and myths are closely-related terms and studied in the same scholarly areas. Fairy tales are tales that involve fantastic forces and beings, while fables often have a moral and involve animals speaking like humans. Myths are defined as traditional stories that "explain a practice, belief or natural phenomenon" (Merriam-Webster Dictionary). Fairy tales, fables and myths are defined according to (aspects of) the content of the tale or their purpose, while the term folktale focuses on the heritage and transmission of the tale. Therefore, a fairy tale, a fable and a myth may be folktales and vice versa. However, we will not delve into questions of exact definitions. Folklorists have published collections of folktales and we will rely on their preselections. The more important aspect is that folktales are tales and therefore comply with all the criteria for narratives. We can expect them to describe sequences of events that are connected so that the story line unfolds and they form a plot in the sense of Forster (1927). Appendix 1 shows the fairy tale "Bearskin" as an example of this.

One of the most prominent collection of folktales is Grimm's fairy tales, published by the brothers Grimm under the title "Kinder- und Hausmärchen". The multi-volume book contains 210 tales and has been published in various editions, the first being in 1812. Andrew Lang's Fairy Books is another well-known collection that contains 437 tales. The books were published in twelve volumes between 1889 and 1910 (Lang, 1889).

4.1.1 Variations and Patterns

Owing to oral tradition, variations on the same plot exist across borders of culture and language. In order to facilitate research on folktales, the Aarne-Thompson-Uther index has been created to classify tales into groups according to "tale types". Tale types have

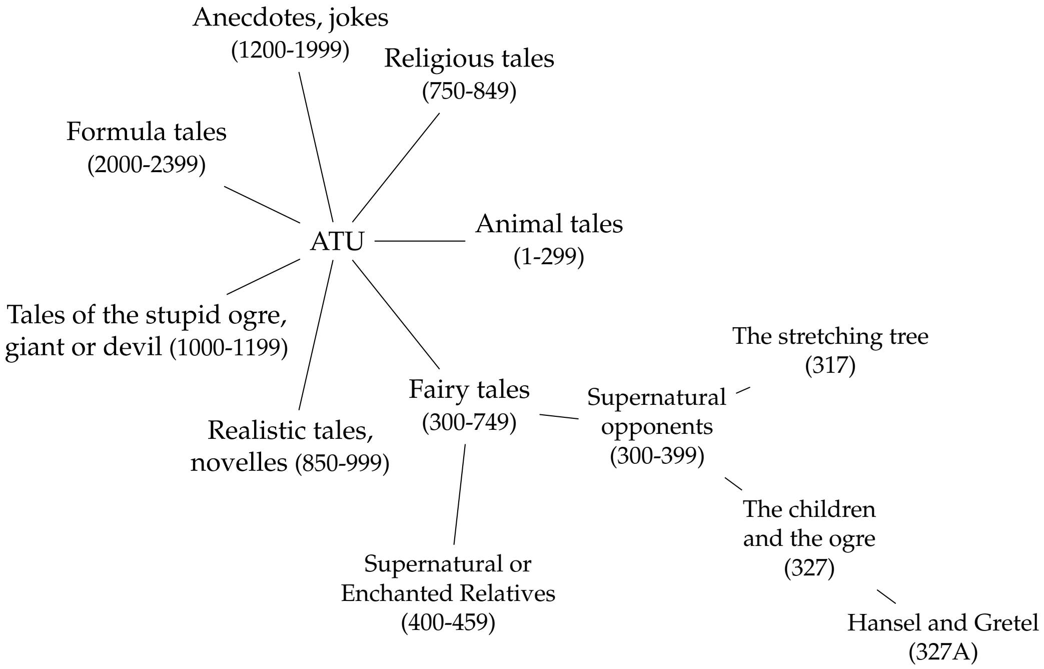


Figure 4.1: Top-level categories in the Aarne-Thompson-Uther index

not been exactly defined. According to their descriptions, a type is defined by some key elements of the story like important actions and prominent characters (see below for an example). The index was first published by Antti Aarne in the early twentieth century. The index has subsequently been extended by Stith Thompson and Hans-Jörg Uther (Uther, 2004) and contains more than 2,500 types. Figure 4.1 shows the top level categories in the index and, as an example, the hierarchy of the index type 327A, *Hansel and Gretel*. Type 327A groups several stories that feature the same elements together:

> The parents abandon their children in the wood. The gingerbread house. The boy fattened; the witch thrown into the oven. . . . The children acquire her treasure.
>
> (Aarne and Thompson, 1961, p. 117)

To our knowledge, we are the first to make use of the ATU classification in a computational narrative analysis setting. As ATU classes are categorized according to their types and types represent story elements, the tales of a given index type necessarily share story elements. However, there are also re-occurring elements in tales classified into different ATU classes. For instance, many tales involve something being forbidden or prohibited (e.g., parents asking their children not to leave the courtyard) and a violation of that command (the children leave the courtyard).

Propp (1958) developed a formal system of thirty-one *event functions* that appear in one hundred tales that he studied. In contrast to a tale type, which makes a statement about a tale as a whole, an event function represents the function of a single event in a tale. Say, for instance, the hero in a tale has to pass a test before he receives a key item needed to defeat the villain. There can be a number of ways this test can be realized (a puzzle, a fight, a riddle, . . .), but the function of this event for the narrative is still that of testing the hero. Some of the event function descriptions indicate various ways of realization as sub-types, as is shown in the following example:

> I. One of the members of a family absents himself from home (Definition: *absentation*. Designation: β.)
>
> 1. The person absenting himself can be a member of the older generation (β^1). . . .
> 2. An intensified form of absentation is represented by the death of parents (β^2). . . .
> 3. Sometimes members of the younger generation absent themselves (β^3). . . .
>
> (Propp, 1958, p. 26)

The above quotation describes an event function that appears during the description of the initial situation. Some of the functions are related to others. For instance, an *interdiction* (γ) is usually followed by a *violation* (δ) of said interdiction.

Similar to the event functions, Propp describes seven character functions: Prototypical roles that appear in tales. Each character function is introduced in a specific event function and may re-appear in others. The *villain*, for instance, appears in the function *villainy* (A), which has a number of sub-types in which the villain causes harm. Later in the tale, he reappears in the functions H (*struggle*, combat between hero and villain) and Pr (*pursuit* of the hero).

Propp further defines a *move* as “any development proceeding from villainy [. . .] through intermediate functions to marriage” (Propp, 1958, p. 92). A single tale may contain multiple moves. He then analyzes the moves in fifty tales from his collection according to the function scheme he developed in detail and published the function strings (the list of event functions as they appear in the tale). The most significant of his findings is that all fifty tales follow the same pattern (shown in 4.1): The first part (A-G) is the same in all tales, then, the story can either take the upper or lower branch or none or both (first the upper and then the lower). The last part (Q-W*) is again the same in all tales (see Appendix 2 for an overview of all event functions).

$$\mathrm{ABC} \uparrow \mathrm{DEFG}\frac{\mathrm{HJIK} \downarrow \mathrm{Pr\text{-}Rs}^0\mathrm{L}}{\mathrm{LMJNK} \downarrow \mathrm{Pr\text{-}Rs}}\mathrm{Q\ Ex\ TUW}^\star \tag{4.1}$$

As an example, Propp published his complete analysis of a single tale. This includes a line-based annotation in which specific lines are associated with event functions (this is governed by typographical constraints). For the remainder of his annotations, however, he did not publish annotations of specific text fragments, but only the function strings.

The Proppian analysis is deliberately formal (given the time at which it was written). It aims to provide a way of finding patterns in tales by comparing the function strings from different tales with each other. Propp explicitly describes his approach as an empirical one.

4.1.2 Computational Narrative Analysis for Folktales

Plot similarities are obviously of interest for folklorists and literary scholars. Both the ATU index and the Proppian framework, however, give little support for the actual, reliable identification of these similarities.

The ATU index does offer a category system but the actual classification is up to the scholar, based on the plot elements he or she identifies. It can be assumed that the identification of such elements is not a straightforward task and that the selection of an ATU type is difficult, even if there were only a few hundred ATU types and not 2,500. To our knowledge, no studies that report any kind of annotator agreement have been published to date. Propp further points out that many tales should actually be classified in multiple classes, as multiple "striking incidents" (Propp, 1958, p. 11), which make up the classes, can occur in a single tale.

Technically solid annotations of Proppian functions have been tried, but with rather poor results. Finlayson (2012) achieves an F_1-agreement of only 0.22 and subsequently redefines the agreement measure so that two annotations are counted as an agreement if they have a substantial overlap (more than half). The F_1-agreement then climbs to 0.71. In Bod et al. (2012), annotators were asked to annotate the Proppian functions directly onto four different tales for which Propp had published a function sequence. Not a single annotator produced the same sequence as Propp, nor did any two of the annotators agree on their function string. Even if the assignment of character functions to characters was given beforehand, the encodings differed vastly. Bod et al. explicitly concluded that it was not worth working on annotations according to the Proppian scheme.

We propose the use of computational linguistics methods to discover plot similarities automatically. Given the annotation issues with Proppian functions and the high variability in the data, we refrain from a fixed inventory of patterns or event functions. Instead, we employ a bottom-up approach, in which similarities are discovered automatically in the texts and can be inspected and interpreted manually.

As a first step, texts are automatically annotated on various linguistic levels and the annotations are linked and integrated. From these annotations, we then extract a sequence of event representations for each document. By applying an alignment algorithm, we can find similar events that appear in different tales. The use of a multifactorial similarity function allows us to go beyond aligning completely equal events (e.g., on the surface level). Instead, we can define exactly how much dissimilarity we allow for an alignment link and which similarity factors are more important than others.

The analysis of the generated alignments reveals areas in tales that have common subplots (indicated by high alignment density) and areas that differ a lot (low align-

Corpus	# documents	# word tokens	# word types	# sentences
All	37	26,551	9,210	1,323
ATU47A	5	3,089	1,070	146
ATU156	6	2,719	1,184	101
ATU225A	7	3,647	1,443	149
ATU333	7	8,496	2,477	501
ATU361	5	5,994	1,950	281
ATU366	3	1,250	504	73
ATU1215	4	1,356	582	72

Table 4.1: Overview of some key characteristics of the folktale corpus

ment density). Common subplots, in turn, are good candidates for plot elements.

4.1.3 Folktale Corpus

We collected a corpus of 38 folktales from seven different ATU index types. Table 4.1 shows an overview of the corpus. The tales were edited by Ashliman (1987) and published online (Ashliman, 1996).

Two main criteria guided the selection of stories: (i) we searched for index types featuring tales that vary in length. Therefore, the tales should also differ in granularity, presumably leading to 1-to-n alignment links. (ii) Secondly, the tales have a relatively clear event sequence (mainly) in temporal order. We omitted tales with long passages of internal monologue or large amounts of direct speech.

The definitions of the ATU index types are as follows:

ATU47A The bear is persuaded to bite the seemingly dead horse's tail. Is dragged off by the horse. The hare asks the destination and laughs till his lip splits.

ATU156 Thorn removed from lion's pawn. In gratitude the lion later rewards the man.

ATU225A Tortoise lets self be carried by eagle. Dropped and eaten.

ATU333 The wolf or other monster devours human beings until all of them are rescued alive from his belly.

ATU361 A soldier bargains with the devil. For seven years he must neither wash nor comb himself. He receives much money. He marries the youngest of three sisters, the two elder of which have made sport of him. The elder sisters hang themselves. The devil: "I got two; you one."

ATU366 A man steals the heart (liver, stomach, clothing) of one who has been hanged. Gives it to his wife to eat. The ghost comes to claim his property and carries off the man.

Coverage	
Types in BNC	98.5%
Nouns in WordNet	97.6%
Verbs in WordNet	99.2%
Verbs in FrameNet	96.2%

Table 4.2: Coverage of resources on the folktale corpus

> ATU1215 Trying to please everyone. . . . Miller blamed when he follows his son on foot; when he takes the son's place on the ass; when he takes the son behind him; and when he puts the son in front of him.
>
> (Aarne and Thompson, 1961)

As the tales were edited for an English-speaking, general public audience, they can be expected to be written in standard language, featuring only small amounts of peculiarities. In order to check this hypothesis, we calculated the coverage of a few key resources on the tales.

Coverage Table 4.2 shows the coverage of several resources with respect to the tales corpus. Almost all of the types present in the folktale corpus are also present in the British National Corpus (BNC). The exceptions are (i) named entities, (ii) non-standard spelling in direct speech (*nough*) or (iii) uncommon compositional word forms (*undraw*, *unclose*).

We also calculated the coverage of WordNet. 97.6% of the nouns and 99.2% of the verbs are indeed present in WordNet. A manual inspection finds that the missing nouns are either named entities or wrongly identified as nouns. Part of speech tagging errors also make up the majority of the missing verbs. In addition, a few rarely used verbs are missing in WordNet: *betake*, *undraw*.

The coverage of FrameNet is lower than WordNet's, but with 96.2% still quite high. Most of the missing verbs are indeed verbs and are simply not present in FrameNet (*horrify*, *swallow*, . . .). This is probably due to the fact that FrameNet has been developed using newspaper corpora.

Timeline Owing to the oral tradition and the focus of fairy tales on children as a target audience, we generally assume that the narrative order (i.e., the order in which events are described in the texts) correlates with the temporal order of the events as they happen in the story. This assumption has been confirmed (Arslan, 2013) with an annotation study in which temporal relations between events were annotated.

We do not, however, assume that this is generally the case for narrative texts. Obviously, many narrative texts contain flashback elements which disrupt the temporal order. Embedded narratives are also important for understanding narrative texts according to Mani (2012) ("narrative levels"). The texts in our collection, however, follow a chronological order.

4.2 Rituals

Research on rituals is an interdisciplinary humanities area that focuses on rituals. Although religious rituals are the most prominent ones, rituals are ubiquitous and can be observed in almost every area of human life, e.g., in politics (inaugurations of monarchs, presidents and chancellors) or culture (tea ceremony, table manners). Rituals also constitute a part of cultural heritage and folklore.

An exact definition of the term 'ritual' is a controversial topic among researchers of rituals and many definitions are intentionally vague. We will not discuss this issue here in much detail. Instead, we pragmatically rely on ritual material that has been published by researchers of rituals (e.g., Gutschow and Michaels, 2005). There are, however, several core assumptions that are made about rituals:

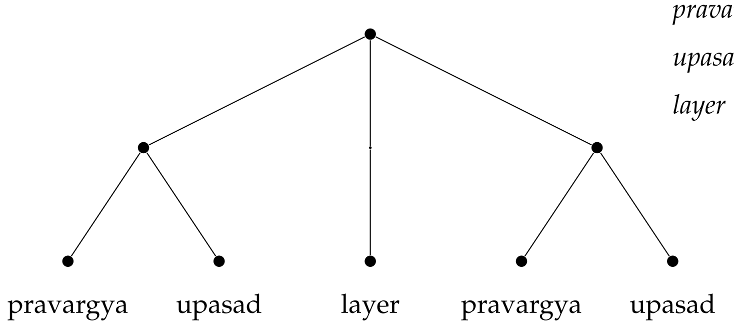
Almost anything may be part of a ritual. Actions of the same action type (e.g., giving money to someone) may be ritual or profane actions. The distinction between ritual and profane actions is not grounded in the actions or action types, but in the context and perception of practitioners. This also means that there is no finite set of ritual actions.

For an outside observer, it may not even be obvious that an action is part of a ritual. However, it can generally be assumed that it is for a practitioner or participant. People participating in rituals usually know that they are participating in a ritual, even if they do not call it one. Similarly, practitioners also have a clear understanding of when the ritual starts and ends (cf. Brosius et al. (2013) for a detailed discussion).

4.2.1 Ritual Grammar

Recent research on ritual has shown that many rituals consist of re-occurring elements that can be exchanged and recombined in a given cultural or religious context. Accordingly, the term "ritual grammar" has been coined to denote structural principles used to combine basic building blocks into more general and complex ritual structures.

The exact nature of the building blocks ("ritual elements") is debated among ritual researchers. Oppitz (1999) argues that mobility and transposability are essential criteria for ritual elements. This refers to the fact that elements of rituals can be reused in other rituals. Michaels (2010) lists six areas that contribute ritual elements: (i) Agency, representing the involvement of those leading a ritual (priests, brahmins), (ii) body, any kind of decoration or use of participants bodies (e.g., putting on jewelry, making certain movements), (iii) language and gestures, for speeches, sayings, prayers and chants, (iv) decoration of the area in which the ritual is taking place, (v) framing, the time slot for a ritual (e.g., on Sundays or at a time determined according to astrological recordings) and (vi) material, special utensils used in the ritual. These aspects can form ritual elements in the context of Hindu rituals, which are studied by Michaels.



pravargya Hot milk is offered to deities

upasad Battle against demons

layer Construction of the layer of an altar

Figure 4.2: (Sub-) structure of a fire ritual according to Staal (1989)

Inspired by generative grammar, Staal (1989) created a rule set describing an old Indian fire ritual. The (context-free) rules can be applied recursively and repeatedly and thus allow for the construction of an infinite number of rituals from a finite set of ritual elements. Staal uses actions with specific participants as basic ritual elements. The embedding rule, for instance, allows ritual elements to be enclosed within other ritual elements: $A \rightarrow BAB$. When combined with the unit formation rule $B \rightarrow DE$, we can describe the construction of an altar and its surrounding events as shown in Figure 4.2.

Lawson and McCauley (2002) focus on the practitioners of rituals and note "striking similarities between speaker-listeners' knowledge of their language and participants' knowledge of their religious ritual systems." They argue that children learn (ritual) rules just the same way they learn language rules. Lawson and McCauley investigated a number of rituals and construct "formation trees", which roughly correspond to syntactic trees for (linguistic) sentence analysis. The formation rules allow for, e.g., repetitions, substitution, fusion and others. Michaels (2012) builds on this inventory of rules in order to describe Newar life cycle rituals. All the analysis in terms of grammar has been performed individually, manually and in a mostly informal way.

4.2.2 Computational Narrative Analysis for Ritual Research

As we have discussed earlier (Section 3.2), a narrative fabula is a chain of events which includes particular actors/objects and unfolds in a given setting. Given this characterization of narratives, it is clear that descriptions of ritual performances can be seen as narrative fabulas (according to Forster's definition, as stories even). Although Michaels (2010) makes no clear distinction between events and participants, the areas he describes as contributing ritual elements can be analyzed in these terms:

(i) Ritual specialists will be mentioned as participants of actions and described accordingly (*the priest*).

(ii) Decorating the body and making movements is expected to be expressed as actions.

(iii) Chants, prayers etc. that have to be uttered during the ritual will appear as semantic role fillers of utterance actions.

(iv) If the decorating is part of the ritual itself, it is expected to be described in terms of actions and participants (who is decorating what) or states. If the decoration is to be performed beforehand, the description of the ritual should describe the setting at the beginning.

(v) The so-called framing of a ritual, i.e., the occasion, time slot or trigger, is also often mentioned at the beginning either as actions or states.

(vi) Specific material and utensils are mentioned in the form of event participants.

It is therefore reasonable to assume that descriptions of rituals can be analyzed using techniques developed for analysis of narrative fabulas. Although this narrative approach does not cover every aspect of ritual analysis, we argue that the question of event/role-structural properties of a ritual ("ritual grammar") can be approached in this way: Areas (i) to (iv) are expressed as actions with participants and can be modeled straightforwardly. (v) and (vi) are partially expressed as statives, which can also be represented as predicate argument structures with role semantic analysis.

One of the key aspects of existing work on computational narrative analysis is their focus on event sequences (e.g., story grammars, narrative schemas, analogical story merging) and as we have seen, several of them aim at detecting typical event sequences from multiple narratives. The general hypothesis for the application of these techniques to descriptions of rituals is that overlapping event sequences across descriptions of the same ritual type (e.g., descriptions of multiple performances of a marriage ritual) show common elements for that type. Say, for instance, the event sequences extracted from two Christian baptizing church services are ⟨*put(water,child)*, *read(priest,text)*, *say(all,our father)*⟩ and ⟨*put(water,child)*, *sing(all,song)*, *say(all,our father)*⟩. From these two sequences, we would then extract the overlapping events (*put(water,child)* and *say(all,our father)*) as common elements for the ritual of baptism. These common elements can then be compared to common elements for other ritual types, in order to identify the elements that are specific to the types. If the above analysis would be done for other Christian rituals, we would identify that *say(all,our father)* is not specific to the ritual of baptism, because it appears in many other rituals as well.

The use of *textual* descriptions of rituals introduces another abstraction layer, in particular if compared to the manual approaches for constructing ritual grammar rules as described above. While ritual structures encoded manually directly encode actions and participants, we work on textual representations of actions and participants. This distinction is important for two reasons: (i) The textual representation may be ambiguous, unclear, incomplete and may contain textual material that is not part of the actual ritual (see below). In this work, our analysis is based solely on the textual material and not, for instance, on any cultural knowledge an annotator might have. As the descriptions of rituals are published to be read by other researchers we generally assume, however, that they include all the crucial actions and existents. (ii) Without using an abstraction layer, rituals would not be accessible for empirical research on a large scale, as they would need to be encoded individually. Instead, textual ritual descriptions are available for rituals from many ritual research contexts, or can be produced relatively easily. Video recordings of rituals may be another abstraction layer, but this poses new questions with regards to image recognition. In the future, motion capturing recordings could be an interesting abstraction layer as well.

We propose the use of similar methods for the analysis of rituals as we are using for the analysis of folktales: Using computational linguistics techniques, we construct an integrated, rich discourse representation for a description of a ritual. This discourse representation also contains a representation of the sequence of events that happen in the ritual. We will use alignment algorithms in order to find similar subsequences across multiple different descriptions of rituals. The common subsequences are then, in turn, good candidates for ritual elements or "building blocks".

4.2.3 Ritual Descriptions Corpus

As a basis for our experiments we collected a corpus consisting of 46 written descriptions of rituals performed by Hindus and Buddhists from Nepal (Table 4.3). The texts were published by Gutschow and Michaels (2005, 2008). The corpus is composed of both *prescriptive* and *descriptive* texts about rituals from the ancient Indian Vedic (saṃskāras) tradition and from the more recent Nepalese tradition.

All descriptions are written in English and were composed by non-native speakers. 18 texts are *prescriptive* descriptions. They are translations of traditional ritual handbooks originally composed either in Sanskrit, Newari, or in a mixture of both languages. Ritual handbooks are used by practitioners to ensure the correct execution of a ritual. The remaining 28 texts were written by researchers who observed the performance of the respective ritual, and thus represent the *descriptive* part of the corpus. As with folktales, the descriptions of rituals also vary in length and granularity. In the Ihi rituals, for instance, the length varies from 123 to 394 sentences. One of the cūḍākaraṇa descriptions is shown in Appendix 3 as an example.

We selected a core corpus of thirteen texts from the 46 descriptions of rituals. The descriptions were selected on the basis of the following criteria: Thematic coherence (we concentrated on four types of initiation rituals and Nepalese Ihi marriage, as can be seen in Table 4.3), frame annotation density (see below), and the percentage of common subsequences of verbs. The experiments in Chapter 6 use this core corpus.

Linguistic Characteristics

Descriptions of ritual feature several special linguistic phenomena on the lexical, syntactic and discourse level. We describe these phenomena in the following, based on Reiter et al. (2011).

Terminology A description of a ritual produced by an expert on rituals (be it a researcher or a practitioner) often contains terminology specific to the cultural context of

Corpus	# documents	# word tokens	# word types	# sentences
All	46	85,997	22,913	4,378
Prescriptive	18	28,125	7,369	1,976
Descriptive	28	57,872	15,544	2,402
Core Corpus	13	26,522	6,513	1,678
anna-prāśana (first food)	2	1,379	511	116
cūḍākaraṇa (hair cut)	3	4,219	1,087	279
Ihi (marriage)	3	15,433	3,262	820
mekhalā-bandhana (dressing)	3	4,430	1,244	368
nāmakaraṇa (name-giving)	2	1,061	409	95

Table 4.3: Overview of key characteristics of the corpus of descriptions of rituals

the ritual. English translation equivalents for these terms often do not exist. In such cases, they typically remain untranslated in the texts (although they are transliterated into Latin characters).

(6) He sweeps the place for the sacrificial fire with *kuśa*.

Kuśa is a Sanskrit term for a kind of grass (*desmostachya bipinnata*) that is very important in these rituals. It is necessary to sweep the ground with *kuśa* and not with any other kind of grass. The term *kuśa* has never been seen by a common, newspaper-trained part of speech tagger nor is it contained in a lexicon of a rule-based grammar.

The descriptions of rituals in the entire corpus contain 3,729 special terms, mostly nouns and proper names (e.g., gods, specific material or actions), corresponding to 0.85 special terms per sentence.

Fixed expressions Most descriptions of rituals contain fixed expressions consisting of multiple words or sentences. These expressions are often prescribed pieces of text which have to be spoken or chanted while a ritual is performed (e.g., *Our Father* in the Christian liturgy).

(7) Salutation to Kubera reciting the mantra *arddha-māsāḥ [. . .]*;

There is no common term in handbooks or scientific literature to refer to such fixed expressions. Sometimes, prayers or chants have a title or name; sometimes, the first few words or the refrain can be given and an expert will know the exact expression from which they are taken. This in turn means that there are multiple ways to refer to the same mantra. It does not make sense to translate the mantras, as their (propositional) meaning is not relevant for the ritual and often not even known to practitioners. However, identifying a mantra is important for the ritual. In total, 850 mantras are mentioned in the corpus.

Imperatives As ritual manuals are often written by and for practitioners, they contain a high percentage of imperative sentences. In a randomly selected sample of (prescriptive) ritual descriptions, we found 20% of the sentences used an imperative construction. The ritual description with the highest amount of imperatives contained over 70% of sentences with imperative constructions. In contrast, only about 2% of the sentences in the British National Corpus (BNC) contain imperatives.

Complex sentence structures Prepositional phrases (PPs) are quite common in the ritual description, as is already apparent from Example 6. Deeply embedded PPs (as in Example 8) are difficult to attach correctly, but appear regularly in the texts.

(8) [. . .] worship of the doors of the house of the worshipper.

The frequency of syntactic coordination and nested sentence structures varies between languages and text types. In Sanskrit, which is the source language of most of our texts, long and nested sentences are very common. This characteristic is also reflected in the texts' translations into English, as translators try to preserve the original character of the text as much as possible and do not aim to produce English sentences which read well.

The occurrence of prepositional phrase attachment along with coordinations as well as sentence embedding poses a challenge for syntactic processing. Example 9 illustrates the interaction of coordination (*italic*) and PP attachments (underlined) in a long sentence.

(9) Beyond the members of the lineage, these visits lead to the paternal aunts of three generations which includes father‘s *and* grandfather‘s paternal aunts *and* their daughters *and* granddaughters, the maternal uncles *and* maternal aunts of their grandmother *as well as* their maternal uncles of three generations.

This leads to a combinatorial explosion of possible analyses and to a real challenge for parse disambiguation. A certain amount of wrong guesses (and therefore noise in the data) has to be expected.

Interpretations Descriptions of rituals that have been published in scientific literature often are not restricted to the ritual performance only. Instead, the factual description is often interwoven with comments or interpretations that help the reader understand the ritual.

(10) The involvement of the nephews can be understood as a symbolic action to address those of the following generation who do not belong to the lineage of the deceased.

Example 10 does not describe an event which happens during the ritual, but a scientific interpretation of it. Although it is possible to represent such sentences in terms of predicate argument structures, they represent a different level of information that does not belong to the ritual itself.

Timeline Because most of the descriptions of rituals are written as manuals, they describe the events in a temporal order. This assumption has been confirmed (Arslan, 2013) by an annotation study in which temporal relations between events were annotated. The annotator almost exclusively annotated *before* relations, indicating the same order in the text as in the rituals.

4.3 Discussion

In this chapter, we have introduced two scholarly areas that deal with cultural heritage: folkloristics and research of rituals. Both textual sources — folktales and descriptions of rituals — are narrative in nature. In addition, the sequences of events and their participants play a major role in the respective areas, because a common goal is the identification of core elements for types. Similarities and variances across texts can be used to highlight these elements.

Another common feature of research of folklore and ritual is that they traditionally not had the means to undertake empirical research on a large scale. The research questions, however, would suggest such approaches. Identifying plot patterns in tales and establishing a ritual grammar presupposes the aggregation of multiple — many — different texts or data sources. This is a challenge for traditional, hermeneutic approaches, for several reasons.

Going over large data sets takes considerable time and resources. In order to detect commonalities in large collections of tales or descriptions of rituals manually and consistently, including going over the same document multiple times, a researcher would need to devote a significant portion of his or her life to a single study, which is just not feasible. This is also acknowledged within literary science. Moretti (2000) describes this as a reason why the canon of literary works that is studied is so small: "you invest so much in individual texts only if you think that very few of them really matter" (Moretti, 2000, p. 57). Following Moretti, this also causes Western-centric view on literature, because most researchers do not do comparative literature studies on a global level.

From our point of view, it is doubtful whether a traditional study of large amounts of data sources can be carried out consistently. If such studies are seen as a kind of annotation, a high intra-annotator agreement is of major importance, i.e., the agreement of the same annotator at different times. But although intra-annotator agreement is generally higher than inter-annotator agreement (e.g., Burchardt et al., 2009; Voormann and Gut, 2008), it is far from perfect. In general, maintaining consistency in large annotation projects is a difficult task independent of the number of annotators, but particularly if such a project runs for a long time. At the very least, the means to reliably detect inconsistencies would need to be made available. In other words; having a single annotator (or researcher) does not make annotations automatically consistent and in hermeneutic studies, inconsistencies are almost impossible to detect.

To summarize, we argue that for research on both folklore and ritual, large-scale empirical approaches need to be explored. This is not to say that empirical research makes traditional approaches superfluous, but, empirical research can support traditional research approaches by offering researchers new views on their data or aggregating them for targeted, manual inspection.

In order to conduct experiments in this direction, we established two corpora, one containing folktales and one containing descriptions of ritual. An inspection of a number of linguistic properties of both revealed that the corpus containing descriptions of rituals had many peculiarities that needed to be addressed for linguistic preprocessing. Both corpora are sub-classified according to events they describe. If other aspects were to prove interesting, the corpus could reflect other variations, e.g., different cultures or eras.

These corpora will be automatically annotated on various linguistic levels in order to create machine-readable discourse representations. Sequences of events and participants in them can then be extracted from these representations. Aligning event sequences according to their semantic similarity allows for the detection of similarities across multiple representations. Comparing similarities found across multiple types in turn allows for the identification of elements that are specific to a certain type.

5 Automatic Semantic Annotation and Domain Adaptation

In this chapter, we will describe the technical architecture of the linguistic processing pipeline in Section 5.1 and the domain adaptation techniques we employed for the ritual domain in Section 5.2. As the domain issues as well as the linguistic annotation levels are quite different, the adaptation techniques do not follow a single paradigm. Instead, each linguistic component is adapted individually. This, in turn, makes a modular processing architecture very important, because components can be adapted in isolation and inserted into the pipeline easily. We will summarize the improvements we have achieved for the linguistic processing of the descriptions of rituals in Section 5.3.

Some of the experiments on domain adaptation have been published before: Part of speech tagging and chunking in Reiter et al. (2011), word sense disambiguation and coreference resolution in Frank et al. (2012). Our approach on the adaptation of dependency parsing and semantic role labeling has not been published before. Adaptations and experiments for coreference resolution have been done by Thomas Bögel, one of the research assistants in the research project on rituals.

We did not perform adaptation of processing tools or resources for the folktales corpus, as the language used in these tales is relatively close to newspaper English (cf. Section 4.1.3 on corpus characteristics).

5.1 System Architecture

NLP processing is done in a single, integrated pipeline. We are using UIMA (Apache Software Foundation, 2014) as a pipeline framework. UIMA prescribes clearly defined interfaces between components, thus enforcing modularization and also making it straightforward. UIMA data structures can be im- and exported using an XML-based file format. Therefore, parts of the pipeline (be it a single or a few components) can also be run individually, by reading from and writing into the XML data format. This is very useful for the development process.

The processing pipeline works by reading in the texts (various importers can be plugged in), processing it in a predefined order and printing out results in a defined format (again, various exporters can be used to export to different formats). Table 5.1 lists the components we have included in the pipeline, Table 5.2 lists package versions and URLs. As the data structures used in UIMA use character positions to indicate begin and end of an annotation (stand-off), new components can be integrated easily.

The only exception to the full integration is the word sense disambiguation component. Instead of calling the UKB program from within the pipeline, the disambiguation

Task	Package	Reference
Sentence splitting	MorphAdorner	
Tokenization	OpenNLP	
Part of speech tagging	OpenNLP	
Chunking	OpenNLP	
Word sense disambiguation	UKB	Agirre and Soroa (2009)
Dependency parsing	Mate	Bohnet (2010)
Coreference resolution	BART	Versley et al. (2008)
Semantic role labeling	Semafor	Das et al. (2010)

Table 5.1: Components used in our preprocessing pipeline

Package	Version	URL
MorphAdorner	1.0	`http://morphadorner.northwestern.edu`
OpenNLP	1.4.3	`http://opennlp.apache.org`
UKB	0.1.6	`http://ixa2.si.ehu.es/ukb/`
Mate	52LX2⋆	`https://code.google.com/p/mate-tools/`
BART	1.0	`http://bart-coref.org`
Semafor		`http://www.ark.cs.cmu.edu/SEMAFOR/`

⋆unreleased

Table 5.2: Package versions and URLs

is done beforehand. Results from UKB are then imported into the pipeline and stored in appropriate data structures.

5.1.1 Import

We are using two different ways of importing textual data. For the import of folktales, we simply read in plain text files. As the descriptions of rituals undergo some preparations before the processing starts (see below), they have to be treated differently. The descriptions are collected in a wiki. This allows the researchers of rituals to edit and prepare them. A UIMA component then uses XML-RPC in order to retrieve the wiki pages directly.

5.1.2 Export into Discourse Representations

The goal of the preprocessing architecture is to create a fully connected discourse representation for each document that contains the semantic representation of events and characters and all the linguistic annotations that have been generated. Figure 5.1 shows a class diagram for the most important annotation types. The diagram shows both the types and the relations between them, small numbers indicating multiplicity of the re-

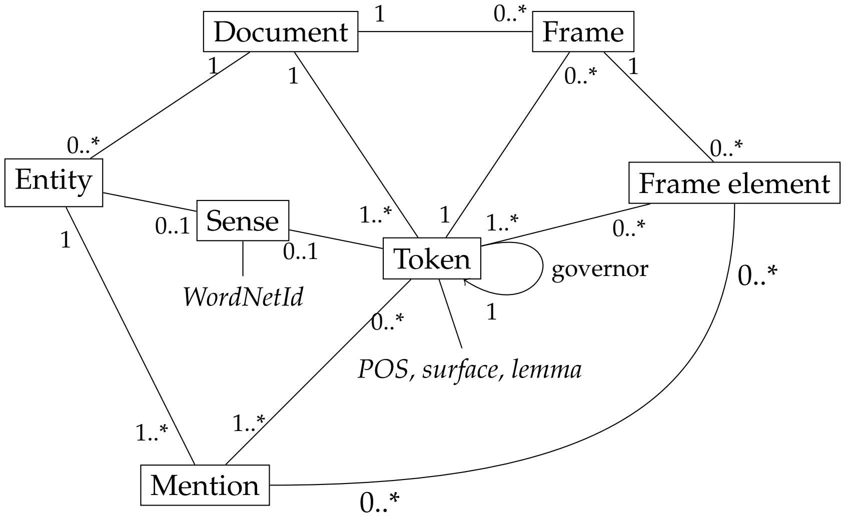


Figure 5.1: Class diagram for discourse representation

lations between annotation types. Most of which are bidirectional. The representation also contains meta data that is not shown in the figure.

Linking annotations

Character-based annotations, as used in UIMA, make integration of different components straightforward. In order to make use of annotations, in particular if they come from different levels of annotation, they need to be linked. In particular, we link mentions with frame element fillers and vice versa. Mention detection and role identification are, due to the modular architecture of the preprocessing pipeline, performed by different components and the annotated spans may differ. However, in a sentence like (11), *the patron* is (ideally) marked as a mention of an entity as well as the filler of a semantic role for the saying event.

(11) The patron says the yathā vihitaṁ karma kuru.

Our algorithm for linking frame element annotations with mentions first checks whether the boundaries of the annotation objects match exactly. If they do, the two annotations are linked. If they do not match exactly, we search for the syntactic head within both annotation objects. This is done robustly by searching for the token that is governed by a token outside of the span.

For a single frame element, there may be multiple linked mentions and vice versa. This is due to the fact that the same span of characters may be annotated as multiple frame elements (of different frames) and at the same time, mention annotations may be coordinated (in, e.g., *hot and/or cold water*, the entire phrase is annotated as a mention as well as *cold water* alone).

Phenomenon	Adaptation step
Terminology Fixed expressions	Marked during input, adapting WSD, CR and SRL
Imperatives	Retraining part of speech tagger
Complex sentence structures	Retraining dependency parser, semantic role labeling

Table 5.3: Ritual domain phenomena and how they are addressed

5.1.3 XML Export

The output format that we finally export represents the discourse representation in XML and is specifically designed for our purposes. All later experiments (Chapter 6) read data from this format. To ensure technical correctness and stability, we use XML schema to validate the exported files. Appendix 5 shows both the XML schema definition and an excerpt of an XML file in this format. Each discourse representation for a document is fully contained within a single XML file. Links between annotation objects are stored using document-wise unique identifiers. By concatenating them with the document identifier, a globally unique identifier can be created.

5.2 Adaptation to the Ritual Domain

In this section, we will describe how we adapted the linguistic preprocessing components to the ritual domain. For most components, we make use of the Wall Street Journal as source domain data set (D_s) and a few annotated descriptions of rituals as target domain data set (D_t). Table 5.3 shows the most prominent linguistic characteristics we described in Section 4.2.3 and how we address them. We did not address interpretative sentences in the descriptions of rituals.

In Section 3.1, we have discussed a lot of different statistical techniques for adapting supervised NLP tools. We will explore one of them for part of speech tagging and chunking (feature space augmentation). However, we focus our work on adaptation techniques that can be employed in digital humanities projects without modification of the source code of training and application programs.

5.2.1 Input Preparation

As the descriptions of rituals contain a lot of foreign words and we expected them to be an issue for automatic processing, we devised a way to handle them. During the input and text collection phase, all foreign words have been annotated with a special markup that also contained circumscriptions in English. The UIMA importer replaces the foreign words by their English circumscription and adds the original term as a UIMA annotation. After the preprocessing is done, the export component replaces them back to the original.

Name	Description	# sentences	# tokens/sentence
WSJ	The Wall Street Journal	47,861	24
RIT	Descriptions of Rituals	532	19

Table 5.4: Data sets for part of speech tagging and chunking

A special kind of foreign words in the ritual domain are fixed expressions like mantras or prayers. Mantras are not directly translatable (even for practitioners) and should therefore be treated differently from other foreign words. Mantras (and chants, hymns, prayers) are all replaced by the indexed word mantra (or chant, hymn, prayer), such that we can reinsert them later and they do not harm the linguistic preprocessing.

5.2.2 Part of Speech Tagging and Chunking

As we aim at a culture- and source language independent framework, we decided to use a statistical part of speech tagger and chunker, that can be trained on specific corpora. Large amounts of training material for both labeling tasks are available from other domains, and the annotation of small amounts of data from the domain of rituals is feasible.

We experimented with two different adaptation techniques: (i) Retraining on mixed data sets makes use of the training procedures in original, but modifies the training data set. (ii) Feature space augmentation uses the technique proposed by Daumé III (2007). This technique also mixes different data sets but in addition modifies the feature space so that in addition to the shared feature space, each domain is represented in its own space. See Section 3.1.1 for details.

Data Sets

As a target domain data set, we manually annotated 532 sentences of the descriptions of rituals with part of speech tags and chunks, using the Penn Treebank tag set. The annotations has been performed in parallel by two annotators. Differences have been adjudicated by the author of this thesis.

We chose the Wall Street Journal as a source domain data set, because it features compatible part of speech and chunk annotations and is reasonably large. For the extraction of chunks from the Penn Treebank we made use of the CoNLL 2000 scripts Buchholz (2000). They were also used for the evaluation of the chunker.

For the marking of chunks, we used a modified version of the CoNLL 2000 style of marking chunks (Sang and Buchholz, 2000): The beginning of PP chunks is marked with `B-PP` as usual. All tokens covered by the PP that are contained in a further embedded NP are marked with a complex chunk tag, for example: `B-NP/I-PP`. This way, we can encode embedded structures in chunks to a certain extent.

Table 5.4 shows an overview of the data sources. We used 10-fold cross-validation to evaluate the performance of the techniques. In cases of training on mixed corpus types

Name	Description	# training sentences (one fold)
WSJ + RIT	Union	48,331
WSJ + RIT ↑	over-sampling RIT	106,955
WSJ ↓ + RIT	under-sampling WSJ	939
WSJ × RIT	Augmented feature space (Daumé III, 2007)	48,331
WSJ × RIT ↑	over-sampling RIT	106,955
WSJ ↓ × RIT	under-sampling WSJ	939

Table 5.5: Training sets for part of speech tagging and chunking

(see below), we "folded" the ritual corpus before mixing it with the Wall Street Journal data. This way, we make sure that our test data did not include any non-ritual data.

Experiments

Table 5.5 shows the different data sets and the sizes of one (average) training fold. WSJ + RIT is a simple union of the two sets. As the sizes of the two data sets differ vastly, we also experimented with equally sized corpora, by use of over- and undersampling. WSJ + RIT ↑ represents the union of the WSJ with the over-sampled RIT corpus, WSJ ↓ + RIT stands for the union of the under-sampled WSJ corpus with the RIT corpus. The data set WSJ × RIT was produced by augmenting the feature space along the lines of the work in Daumé III (2007) (see Section 3.1.1).

Results and Discussion

Part of speech tagging Table 5.6 lists the results obtained by training the part of speech tagger on different data sets. The differences between the best three results are not significant (marked in bold). We use the model trained on the WSJ data set only, i.e., without any domain adaptation, as a baseline. Its performance is 90.9% accuracy.

If RIT is used as (small) training set, the part of speech tagger achieves a performance of 94.82%. Training on the union of RIT and WSJ yields an increase in performance (95.72%) compared to RIT. Balancing the training sets again increases the performance if the ritual data is oversampled (resulting in a very large training set). If the WSJ data is under-sampled, performance decreases compared to the unbalanced union. Augmenting the feature space yields minor improvements, even if the training data is unbalanced. The best performing model is trained on WSJ × RIT, while WSJ × RIT ↑ performs similarly (and the difference between the two is statistically insignificant). The small data set, WSJ ↓ × RIT, achieves less performance than a large and balanced, but not augmented data set (WSJ + RIT ↑). The improvement of the feature space augmentation compared to the best performing non-augmented model is also not statistically significant.

Training data	Accuracy
WSJ	90.90
RIT	94.82
WSJ + RIT	95.72
WSJ + RIT ↑	**96.23**
WSJ ↓ + RIT	95.25
WSJ × RIT	**96.86**
WSJ × RIT ↑	**96.85**
WSJ ↓ × RIT	95.92

Table 5.6: Results for adaptation of part of speech tagging

Training data	Precision	Recall	$F_{\beta=1}$
WSJ	86.3	87.0	86.6
RIT	85.5	86.0	85.7
WSJ + RIT	86.3	87.0	86.6
WSJ + RIT ↑	87.7	88.5	88.1
WSJ ↓ + RIT	86.9	79.7	83.1
WSJ × RIT	74.0	74.9	74.4
WSJ × RIT ↑	81.0	81.5	81.3
WSJ ↓ × RIT	74.8	71.8	73.3

Table 5.7: Results for adaptation of chunking

Chunking Table 5.7 shows the results of the chunking models trained on the different data sets. Again, we use a model trained on the Wall Street Journal as baseline (WSJ). This model achieves an f-score of 86.6. The model trained on the ritual data (RIT) performs slightly lower, achieving an f-score of 85.7. Training the model on the simple union (WSJ + RIT), does not increase the performance compared to the baseline. However, if we oversample the ritual data and thus balance the training data (WSJ + RIT ↑), we achieve a minor improvement in f-score. Undersampling the WSJ data decreases the performance. The augmentation of the feature space decreases the performance on all data sets. This is in contrast with the results for part of speech tagging (above). Within the augmented feature space models, we can observe similar tendencies as in the other models: Oversampling improves the performance compared to unbalanced data, while undersampling decreases it.

Augmentation The results of the feature space augmentation technique show no significant improvement over the use of comparably mixed, not augmented feature spaces. We therefore refrain from using this technique in the following experiments, as it often requires rewriting of source code (in particular the feature extraction part).

5.2.3 Dependency Parsing

The default models provided with the Mate parser are trained on the CoNLL data sets and thus produce CoNLL dependency structures. In order to get more meaningful dependency relations, we decided to retrain the parser using Stanford dependencies (Marneffe and Manning, 2008).

Data sets

We use the Penn Treebank (WSJ, sections 1 to 21), converted to Stanford dependencies using the Stanford Core NLP package. Additionally, we add 95 annotated questions and imperatives provided by the parser developers (Stanford NLP Group, 2014). This represents the source domain data set, D_s.

For the ritual domain, we annotated three sets of sentences manually, in total 191 sentences. Two sets (A and B) are complete descriptions of rituals, the sentences in the third set (C) have been selected for their complexity. Two research assistants annotated the sentences, differences have been adjudicated. Most problems were caused by sentences with nonstandard syntax (e.g., sentences without verb). In these cases, we tried to decide on an analysis that most accurately represents the meaning of the sentence.

Experiments

We compare three different settings of the dependency parser. (i) The performance of the unadapted dependency parser running on part of speech tags that have been produced by an unadapted part of speech tagger (no adaptation). The unadapted part of speech tagger uses the default model provided with OpenNLP, trained on the Wall Street Journal. (ii) We use the unadapted dependency parser on adapted part of speech tags (partial adaptation). (iii) In the third setting, we run the adapted dependency parser on adapted part of speech tags (full adaptation). The parser is adapted by using two thirds of the annotated data as additional training data while holding back one third as test data to avoid overfitting.

Results

Table 5.8 shows the results of the experiment. The first column indicates the status of the part of speech tags (adapted or not). The results already improve by using adapted part of speech tags. This is not surprising, but it highlights the “pipeline effect” in a positive way: Improvements in earlier processing stages also improve later processing stages, without any intervention in these stages. Noteworthy is further that the improvements gained by adapting the part of speech tags differ between the documents: The improvement on document B is very small (+1.3 LAS), but much larger on C: +13.1 LAS. The (averaged) gain by using adapted part of speech tags is +8.3 LAS.

Part of speech	Training	Test	LAS	UAS
Unadapted	D_s	A	73.5	76.1
	D_s	B	76.0	79.1
	D_s	C	70.4	75.1
	$\varnothing$		72.8	76.4
Adapted	D_s	A	80.8	82.9
	D_s	B	77.3	79.6
	D_s	C	83.5	86.8
	$\varnothing$		81.1	83.8
	$D_s \cup B \cup C$	A	83.9	84.9
	$D_s \cup A \cup C$	B	79.5	82.3
	$D_s \cup A \cup B$	C	85.7	88.7
	$\varnothing$		83.6	85.9

Table 5.8: Results for adaptation of dependency parsing

The results with adapted part of speech tags and unadapted parsing are between 77% and 83% labeled attachment accuracy (avg. 81.1%), which is not far below the state of the art for labeled attachment accuracy on Stanford dependencies (Cer et al., 2010). If the parser is domain adapted (i.e., the training set contains some amount of domain data) the performance improves by about 2.5 LAS on average. Unlabeled attachment score is about 2 percentage points higher in the adapted scenarios.

5.2.4 Word Sense Disambiguation

We use UKB for word sense disambiguation. UKB works by applying the PageRank algorithm on the WordNet concept graph. As highlighted in Section 3.1.2, there are two obvious ways to adapt UKB to new domains: By adapting WordNet or by adapting the algorithm, in particular its initialization. Given the existence of a sense-annotated corpus that could be employed, we chose to adapt the WordNet database.

Data set

To build a gold standard for testing UKB's performance, we randomly chose 50 sentences from all descriptions of rituals. These sentences were annotated independently by two annotators with word senses from WordNet 2.0. Both annotators have a computational linguistics background. Differences between the two annotations have been adjudicated.[1] This resulted in 462 annotated nouns, verbs, adjectives and adverbs, forming our gold standard for WSD.

[1] In two cases WordNet 2.0 did not contain appropriate concepts for annotation: "*bel* fruit" (Sanskrit *bilva*; a fruit used for worshipping Śiva) and "*block* print". These words were left unannotated.

		MFS	$\text{UKB}_{\text{WN 2.0}}$	$\text{UKB}_{\text{+rit-node}}$
Nouns	Coverage	94.5	93.3	93.3
	Precision	59.8	60.2	**64.1**
	Recall	**60.0**	53.7	57.3
	F-Score	59.9	56.8	**60.5**
Adjectives	Coverage	88.4	86.9	86.9
	Precision	48.3	**51.2**	49.8
	Recall	**49.3**	**49.3**	47.8
	F-Score	48.8	**50.2**	48.8
All Words	Coverage	94.3	93.1	93.1
	Precision	53.9	54.2	**56.4**
	Recall	**54.5**	49.9	51.8
	F-Score	**54.2**	51.9	54.0

Table 5.9: Results for adaptation of word sense disambiguation

Evaluation measure

We assessed the performance of UKB using precision and recall as evaluation metrics, calculated for individual word types and micro-averaged over all types. As the semantic annotation of verbs will be mainly covered by FrameNet annotations, we specifically report on the performance of WordNet sense disambiguation for nouns and adjectives, next to performance on all words. The word sense disambiguation system selects candidate synsets based on the part of speech tags provided by the domain-adapted tagger.

Domain adaptation for word sense disambiguation

In order to adapt UKB to the ritual domain, we enriched the WordNet database with domain-specific sense information. We acquired senses that may be characteristic for the ritual domain from a Digital Corpus of Sanskrit (Hellwig, 2010). This corpus is designed as a general-purpose philological resource that covers Sanskrit texts from 500 BCE until 1900 CE without any special focus on the ritual domain. In this corpus, approximately 400,000 tokens had been manually annotated with word senses from WordNet 2.0. Using this annotated corpus for domain sense acquisition was motivated by the supposition that even general passages from Sanskrit literature may contain a significant amount of senses that are relevant for the ritual domain.

We linked all 3,294 word senses that were annotated in this corpus to a newly introduced non-lexicalized pseudo-synset `rit-topic`. As UKB calculates the page rank between sense-related words in the WordNet database, introducing this node increases the chances that senses specific for Newar culture receive a higher rank.

Results

The performance results for different system configurations are summarized in Table 5.9. We assigned the most frequent sense (MFS) from WordNet 2.0 as a baseline. This baseline achieves a precision of 53.9% and a recall of 54.5% for all words. For 5.7% of the tokens, the baseline implementation does not return a word sense. This loss in coverage is mainly caused by erroneous part of speech assignments.

We first tested the performance of UKB 0.1.6 using standard WordNet (2.0). The system achieves a precision of 54.2% and a recall of 49.9% (for all words) and thus performs below the MFS baseline (the loss in recall outranks the gain in precision), which is not unusual for unsupervised WSD systems. The coverage drops by a small amount to 93.1%.

As seen in Table 5.9, linking domain-related senses to a pseudo-synset results in an improvement of 2.2 points in precision and 1.9 points in recall for all words, when compared to $\text{UKB}_{\text{WN2.0}}$. Moreover, the domain-adapted UKB system now closely matches the MFS baseline in F-Score. Note further that for nouns the domain-adapted WSD system obtains the best results (P: 64.1%, F: 60.5), and outperforms the MFS baseline in terms of precision (+4.3) and f-score (+0.6), with only a slight loss in recall (57.3%; -2.7) and coverage remaining stable.

5.2.5 Semantic Role Labeling

Semafor (Das et al., 2010) is a supervised system for FrameNet frame parsing and semantic role labeling that has achieved high performance numbers for both tasks (exact frame matching on predicted targets: 61.4 F_1, fully automatic argument detection: 46.5 F_1). We used Semafor as a system and decided to retrain on mixed data sets.

Data set

Frame annotations have been performed by correcting automatically produced annotations. First, the original model of Semafor (trained on FrameNet data) was used to assign frames in unannotated descriptions. The assigned frames were checked by two annotators, and differences were adjudicated by a supervisor. In a second step, semantic roles were assigned manually to the adjudicated frames by two annotators, and were again checked for consistency by the supervisor.

We added two ritual specific frames to the FrameNet hierarchy because the applicable frames in FrameNet were not able to capture the relevant meaning aspects for rituals or too broad in their meaning.

The original frame FILLING describes both the filling of a container and the covering of an area. After careful inspection of the description, we decided that using this frame would introduce too much of an abstraction. We therefore created the frame FILLING_RITUALLY, specifically for the filling of containers. In terms of hierarchy, this new frame easily inherits from the original FILLING frame.

Training	Coverage	Precision	Recall	F-Score
FN	70.94	40.25	28.67	33.48
RIT	94.65	96.52	91.36	93.86
FN ∪ RIT	97.14	98.61	95.79	97.18
FN↓ ∪ RIT	96.24	96.19	92.57	94.34

Table 5.10: Results for adaptation of frame labeling

Acts of saluting someone or something seem to be not covered in FrameNet. Neither *to salute* nor *to greet* are included as a lexical unit in FrameNet. Because salutations are important for rituals, we added the frame SALUTE_RITUALLY to inherit the frame STATEMENT.

Depending on the complexity and the ambiguity of a frame, we observed an inter-annotator agreement between $\kappa = 0.619$ (frame MANIPULATION) and $\kappa = 1.0$ (frame CUTTING) for frame annotation. For role annotation, we observed a global $\kappa = 0.469$, which indicates rather low agreement. However, a closer look at the data reveals that 89.4% of the differences in role annotations occur when one annotator annotates a role that the other annotator does not recognize.

Using this double annotation approach, we built up a domain corpus of manually checked frame semantic annotations that contains 1540 frames of 15 different types and 3197 roles of 95 different types.

Experiments

We are adapting Semafor to the ritual domain by retraining its models on various data sets. FN represents the original annotated FrameNet corpus, RIT is the manually annotated data set consisting of descriptions of rituals. FN ∪ RIT stands for the union of the two and FN↓ ∪ RIT for the union of the under-sampled FrameNet corpus with the descriptions of rituals corpus. We restrict the evaluation of frame assignment to cases in which the frame target lemma is included in the training set. Technical issues with the large data set size prevented us from evaluating FN ∪ RIT ↑.

Table 5.10 shows the results for the frame labeling task. Compared to the results of using a standard model, we can achieve an improvement of 63.7 f-score by using the union data set. It is noteworthy that the union data set achieves the highest results, indicating that Semafor benefits from both in-domain and out-of-domain training data.

Error analysis of the performance of the FN model for *frame labeling* shows that the performance varies strongly depending on the frames. Semafor performs poorly with frames that carry culture-specific notions or are evoked by rare lexemes. For the frame TEXT_CREATION, for instance, Semafor yields R: 0.21, P: 8.33 and F: 0.41, because it labels target words such as "chant" consistently with the frame COMMUNICATION_MANNER, while we annotated the frame TEXT_CREATION in these cases[2]. The

[2] It is questionable whether the utterance of a mantra in a ritual is a communication event, as it usually

Training	Precision	Recall	F-Score
FN	17.75	18.04	17.89
RIT	72.88	75.58	74.21
FN ∪ RIT	86.20	86.79	86.49
FN↓ ∪ RIT	72.41	74.91	73.64

Table 5.11: Results for adaptation of frame element labeling

high number of unrecognized instances can be explained by the fact that nouns such as "mantra", which are missing in FrameNet, are annotated manually with the frame TEXT_CREATION. On the other hand, we observe good accuracy for less specialized frames such as PLACING (R: 77.49, P: 60.17, F: 67.74).

The results for role labeling behave similarly (Table 5.11): FN ∪ RIT achieves by far the best performance, with an improvement of 68.6 f-score compared to using only FN.

The evaluation of *semantic roles* was restricted to the roles of those frames that were annotated correctly by Semafor. On these 1268 roles, Semafor achieved P: 73.57, R: 77.29 and F: 75.38, allowing both partial and perfect overlap of spans; P: 70.35, R: 73.90, F: 72.08 if restricted to perfect match.[3] As major error sources we identified non-local roles and non-core roles that are missing in Semafor's output, domain specific vocabulary of our texts, and syntactic peculiarities such as numerous imperative constructions. On the whole, we are confident that system annotations for frames and roles can be improved by retraining Semafor on our labeled domain data.

5.2.6 Coreference Resolution

The coreference resolution system BART (Versley et al., 2008) is a supervised system that implements the methodology and the feature set presented in Soon et al. (2001). BART is tightly integrated with its own preprocessing pipeline. This makes domain adaptation difficult. Given extremely poor results when using BART as off-the-shelf coreference resolver, the need for domain adaptation was obvious, because coreferences are crucial in order to represent events in narratives. Due to the small number of documents in our descriptions of rituals corpus, a retraining approach to domain adaptation is unreasonable. Instead, we employed several other ways of adapting BART:

(i) To reduce noise, we adapted BART's integrated preprocessing pipeline, using our own components for part of speech tagging and chunking. After comparative evaluation with mixed results (see below) we chose to use BART's original parsing pipeline with our own tokenizer. Two further enhancements are used to tailor the system to the ritual domain. (ii) After mention detection, a WordNet lookup filters out mentions of specific semantic classes. This allows us to concentrate on the most important and

lacks an addressee.

[3] Precision rises to 73.90/77.29 (perfect/partial match) if the evaluation is restricted to roles contained in the gold standard. Precision could be slightly underestimated due to a number of roles (64) in Semafor's output that are not annotated in the gold standard, but could still be correct.

	MUC			B^3		
	P	R	F	P	R	F
Gold standard chunks	38.88	50.9	44.09	26.96	39.39	32.01
Adapted Pipeline (chunks)	37.33	50.9	43.07	25.78	40.08	31.38
BART Pipeline (Stanford Parser)	38.27	56.36	45.58	23.48	44.78	30.81

Table 5.12: Results for coreference resolution with domain-adapted chunking and full parsing components

most frequent entity types: persons and supernatural beings such as gods (as opposed to inanimate objects). Moreover, (iii) we included domain-specific knowledge to improve the predictions of BART's semantic agreement features: We extended BART's internal database for names and its procedures with a new category for gods. We also added gender information for items frequently occurring in ritual texts to the existing knowledge databases.

Evaluation

We evaluated BART's performance on manually annotated gold standards using the standard MUC (Vilain et al., 1995) and B^3 (Bagga and Baldwin, 1998) measures as evaluation metrics.

(i) Preprocessing We tested different pipeline architectures, using our own domain-adapted chunker (adapted pipeline) in contrast to BART's pipeline including full parsing with the Stanford parser. We further compared the results obtained using our domain-adapted chunker to gold chunk information (cf. Table 5.12). This evaluation uses a gold standard sub-corpus, a single ritual text, consisting of 40 mentions.

Using chunks provided by the adapted pipeline almost reaches the performance on gold chunks. In general, BART operating on chunks achieves better precision according to the B^3 measure, which is the stricter measure for evaluating entity chains, while the BART pipeline performs better according to MUC. But given the small differences and evaluation data sets, we currently chose to stick to the BART pipeline.

(ii, iii) Sense restrictions and domain knowledge. In further experiments, we evaluated the two domain-specific adaptions discussed above: (ii) restricting coreference resolution to entity subtypes, and (iii) extending BART's semantic knowledge by adding gender information and semantic categories for frequently occurring terms. Here, we used an extended gold standard (3 ritual texts) consisting of 344 mentions. In this experimental set-up, we used the BART pipeline with our own tokenization module.

Table 5.13 shows high performance improvements for *sense restriction* to the entity types *person* and *god*. This holds both for the standard gender model of BART (upper part) and the domain-adapted model (lower part). In both scenarios we observe high

		MUC			B^3		
		P	R	F	P	R	F
Standard	(all)	37.68	59.77	46.22	28.79	46.28	35.5
	(person only)	63.44	57.86	60.52	47.22	38.39	42.35
	(object only)	25.64	49.5	33.78	23.89	48.88	32.1
Domain gender model	(all)	44.62	62.06	51.92	34.23	47.86	39.92
	(person only)	65.21	56.6	60.6	49.04	33.36	39.7
	(object only)	25.64	49.5	33.78	23.89	48.88	32.1

Table 5.13: Results for adaptation of coreference resolution with entity type restrictions and a domain-adapted gender database

gains in precision and f-score, with losses in recall. This fits well with our main interest in analyzing event chains from rituals, where coreference information for the main actors is of primary importance, and our general interest in achieving high-quality annotations.

For the domain-specific enhancements to the *gender model*, both recall and precision increase across all metrics when taking all mentions into consideration. However, mentions of category *object* are not affected.[4] Precision of *person* mentions improved substantially at the cost of a decline in recall, yielding better results for both evaluation metrics. Overall, we achieve best precision figures for the *person-restricted domain-adapted gender model*, with a boost of 20.2 points (B^3) and 27.53 points (MUC) when compared to the standard BART model, at comparable f-scores.

5.3 Summary

In this chapter, we have described our linguistic processing architecture and how we adapted existing tools for linguistic processing to the ritual domain, thus addressing the challenge of uncommon text characteristics (cf. Chapter 2). The architecture is highly modularized and produces a rich, highly connected discourse representation. Although character-based data structures make integration of different components straightforward, the different levels of annotation need to be linked at some point in order to make use of them.

We were able to improve the performance of the linguistic processing tools on the ritual domain substantially by employing various domain adaptation strategies. Table 5.14 summarizes the most important improvements, compared to a non-adapted baseline in each case and the data set sizes we used. The methods we used to achieve these improvements are diverse. For part of speech tagging, chunking, dependency parsing and semantic role labeling, we annotated a small data set from the ritual do-

[4] This is partly explained by the fact that this category is not distinguished by different genders in English, and our focus on the person category when extending the gender database.

Level	Improvement	Domain data set size
Part of speech tagging	+5.3%	532 sentences
Chunk	+1.5 f	
Dependency parsing	+9.5 UAS	191 sentences
Word sense disambiguation (nouns)	+0.6 f	
Semantic role labeling	+68.6 f	1.540 frame instances
Coreference resolution	+6.3 f (MUC)	

Table 5.14: Improvements achieved by adapting linguistic analysis components to the ritual domain

main and retrained statistical models, mixing in the domain data. This worked robustly in this setting and was mostly straightforward to implement. The use of feature space augmentation, as a more complex technique, did not improve the performance, compared to the union of data sets. Both coreference resolution and word sense disambiguation have been adapted in an unsupervised manner, because in both cases the amount of domain data that we would need to annotate was quite large.

In research projects in the area of digital humanities, domain adaptation is usually not the main focus but an instrument to improve processing results. Therefore, there will always be a consideration between effort and expected gain. It is hard to give general conclusions about that, because the main goals – and therefore the need for specific annotations – is very different. However one should keep in mind the pipeline effect: Adaptations on a lower level of linguistic analysis have effects on higher levels. Improving the quality of part of speech tagging, for instance, indirectly influences the quality of all processing stages that build upon part of speech tags. As we have seen for dependency parsing, the improvement gained by adapting part of speech tagging outranks the improvement gained by adapting the dependency parser.

We have also seen that retraining approaches can achieve performance improvements on the same level as more complex approaches discussed in Chapter 3. Developments such as those in the context of the infrastructure project CLARIN-D make retraining approaches available to researchers from humanities: The integration of a web-based processing pipeline (Hinrichs et al., 2010) with an annotation tool (Yimam et al., 2013) and the training of statistical models on the basis of the annotations is currently in development. This will make retraining available as an adaptation technique to many researchers from the humanities, even without deep insight into statistical techniques.

6 Discovering Structural Similarities

In this chapter, we will describe the alignment-based methodology we propose for the discovery of story similarities in large-scale settings. An overview of the general methodology will be given in Section 6.1, along with the experimental tasks that we derive in order to evaluate the performance of the algorithms. In Section 6.2, we describe the alignment algorithms that we employ in order to detect similar events across stories. We will discuss the gold standard and the evaluation methods and measures we are using in Section 6.3. The two experiments we conduct will be discussed, evaluated and analyzed in Section 6.4 and 6.5. In Section 6.6, we will describe an algorithm to detect and rank structural similarities based on event alignments.

6.1 Discovering Story Similarities through Event Alignments

In Chapter 4, we have described how both folkloristics and ritual research can benefit from automatically detected story similarities. Beyond a mere classification of tales, which is done with the ATU index, Propp proposed the use of so-called event functions in order to describe the story line in tales and to detect similar story elements. Researchers of rituals are discussing the existence of structural principles that govern the combination of individual actions into a ritual, because striking similarities in the "story lines" of different types of rituals have been observed.

In both scenarios, a key observation is that similar events appear across stories. In order to assess the similarity of events, we focus on two aspects: (i) The action itself and (ii) the sentient and non-sentient participants. The action itself is expressed as a verb or noun in texts. Participants are described in terms of semantic arguments of the verb or noun.

In both scenarios, the similarity that is sought goes beyond the similarity of individual events. The important finding of Propp was not that heroes fight villains in multiple tales, but that there is a structure in the tales. The events happen in a certain order and this order is similar across tales. Similarly, the striking observation in ritual research was not that the same kind of grass is burnt in different rituals, but that rituals are structurally similar, e.g., that a specific mantra is spoken before the grass is burnt.

We consider two aspects of *structural similarity* between (sub-) sequences of events: (i) The similarity of individual events and (ii) the similarity of the order in which similar events appear in the sequences. Figure 6.1 shows this visually, with arrows indicating the sequence ordering and dashed lines connecting similar events. Although both sequence pairs contain four pairs of similar events, we would consider $\langle A, B\rangle$ as structurally more similar than $\langle C, D\rangle$, because the individual events are appearing in the

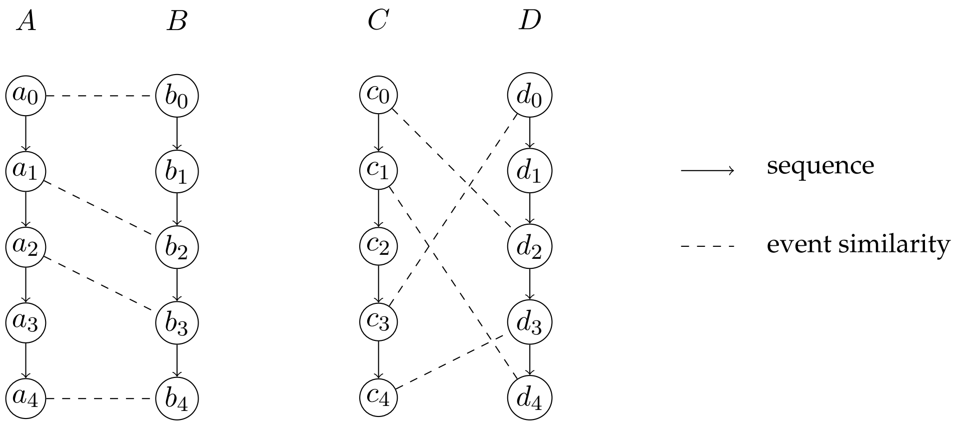


Figure 6.1: Sequences $\langle A, B\rangle$ are structurally more similar than $\langle C, D\rangle$

same order.

In order to operationalize the detection of structural similarities across texts, we are casting this as an alignment task in which similar events across stories are aligned. Alignment algorithms do not align events in isolation, but in their contexts, taking the order of events into account. If we provide lists of events extracted from different documents to an alignment algorithm, the algorithm generates a set of links that denote corresponding events. Consecutive alignment link sequences mark structural similarities.

We will use three different alignment algorithms for aligning events across narrative texts. (i) The first algorithm (Needleman-Wunsch) serves as a baseline algorithm. It has been developed in bioinformatics and has been used in many alignment tasks. The sequence alignment algorithm produces a global, pairwise alignment without crossing links. (ii) The second algorithm is a graph-based clustering algorithm. It has been developed in order to align events in newspaper articles and has not been used before on data from the humanities. It may generate crossing links but is developed for pairwise alignments. (iii) The third algorithm (Bayesian model merging) induces a hidden Markov model (HMM) from multiple sequences. Alignments can be extracted from the HMM. All three algorithms make use of a multifactorial similarity function that we provide in order to assess similarity of individual events.

This operationalization as an alignment task also makes evaluation theoretically straightforward: Automatically produced alignments can be compared against a manually annotated gold standard. In practice, however, event alignment gold standards for our data sets or domains are not directly available and hard to produce. We will therefore use alignment density as a global measure for the similarity of entire stories. Alignment density is defined as the number of linked in relation to the lengths of the sequences. This story similarity, in turn, can be used to induce a clustering of the documents, which can then be compared to existing classifications present in the corpora.

Consequently, we will perform two experiments in order to evaluate the performance of the event alignment algorithms for the detection of story similarities. Table 6.1 shows

	Rituals	Fables
Experiment 1: Gold standard	✓	–
Experiment 2: Cluster Induction	✓	✓

Table 6.1: Experiment overview

an overview of the experiments and in which application scenarios they work. In the first experiment, the outputs of the alignment systems are compared directly to an annotated gold alignment for descriptions of rituals. We evaluate the produced alignments with the Blanc score, a measure introduced for the quality assessment of coreference resolution systems. As we will describe, producing such a gold standard is a difficult task.

Therefore, the second experiment does not rely on such an alignment gold standard. Instead, we use the alignments generated by the algorithms in order to induce a clustering of the input documents. This clustering can then be compared to a previously known clustering of the documents: The ritual descriptions are grouped according to their ritual type, the tales are grouped according to overlaps in their plots (encoded in the ATU index). Both classifications have been described in Chapter 4. We evaluate the cluster quality with the Rand index.

While the clustering induced by the alignment density allows a global view on event-based story similarity, the individual alignments show event similarity on a local and fine-grained level. To support researchers from the humanities with the fine-grained analysis, we will describe a graph-based algorithm that allows targeted inspection. The algorithm ranks events according to their connectivity to another sequence. Based on this score, we can identify regions that are structurally similar across stories.

6.2 Event Alignment Algorithms

This section describes the three alignment algorithms we employ. We will first describe the algorithms, give an example and then highlight their key properties in comparison. As all algorithms make use of a function for measuring semantic similarity of individual events, we will describe the similarity measures at the end of this section.

6.2.1 Sequence Alignment

The Needleman-Wunsch algorithm (Needleman and Wunsch, 1970) works on two input sequences $S = \langle s_1, s_2, \dots s_n \rangle$ and $T = \langle t_1, t_2, \dots t_m \rangle$ over an alphabet $\mathcal{E}$ ($s_i \in \mathcal{E}, 1 \leq i \leq n$ and $t_i \in \mathcal{E}, 1 \leq i \leq m$). It generates a global alignment (i.e., every element in both sequences is either linked or skipped) and an alignment score. The global alignment does not include crossing links, but may contain gaps and mismatches.

The algorithm relies on two functions: A gap cost and a similarity function. The gap cost function $\mathrm{g} : \mathbb{N} \to \mathbb{R}$ assigns a cost for the introduction of gaps. The cost depends on the size of the gap. The similarity function $\mathrm{sim} : \mathcal{E} \times \mathcal{E} \rightarrow \mathbb{R}$ gives a score to the similarity of two sequence elements. Usually, sim assigns a negative score to mismatches (dissimilar sequence elements) and a positive score to similar elements.

The backbone of the algorithm is an $n+1 \times m+1$-matrix M in which n rows represent the elements of sequence A and m columns the elements of sequence B. A cell in the matrix then stands for a link of two sequence elements. Initially, the cell in the top left is filled with a 0 and the first row and the first column are filled according to the gap function. Then, the remainder of the matrix is filled according to Equation (6.1).

$$M[i,j] = \max \begin{cases} M[i-1,j-1] + \mathrm{sim}_{\mathrm{NW}}(a_i,b_j) & \text{Match/Mismatch} \\ \max_{1\leq k\leq i} M[i-k,j] + g(k) & \text{Gap} \\ \max_{1\leq l\leq j} M[i,j-l] + g(l) & \text{Gap} \end{cases} \tag{6.1}$$

The overall alignment score can then be found in the bottom right cell of the matrix. The alignment can be extracted by tracing the individual decisions back through the matrix. A global alignment is achieved if the path goes from the top left element to the bottom right element (therefore, all elements in both sequences are handled).

Originally, the Needleman-Wunsch algorithm has been developed for use in bioinformatics for the alignment of protein or nucleotide sequences. Proteins and nucleotides are represented by upper-case letters and their is a finite set of them. Measuring similarity of proteins and nucleotides is not an issue, as they are either equal or not. In order to incorporate our semantic similarity function into the Needleman-Wunsch algorithm, we scale the values it returns (mismatches should be represented by negative numbers). Values above the threshold t are scaled to $[1,2]$ and values below t to $[-1,0]$. We use $g(n) = -n$ as gap cost function (i.e., introducing a gap costs 1 point).

Example

As an example, we will align the two sequences $S = \langle a,b,a\rangle$ and $T = \langle b,a\rangle$. We assume identity as a similarity function, such that $\mathrm{sim}(a,a) = 1$ and $\mathrm{sim}(a,b) = 0$ and a threshold of 0.5, such that the scaled values are -1 (for mismatches) and 2 (for matches). Initially, the matrix is filled as shown in 6.2.

$$M_0 = \begin{bmatrix} 0 & -1 & -2 & -3 \\ -1 & & & \\ -2 & & & \end{bmatrix} \tag{6.2}$$

$$M[1,1] = \max \begin{cases} M[0,0] + (-1) & \text{Aligning } a \text{ and } b \\ M[0,1] + (-1) & \text{Gap in } S \\ M[1,0] + (-1) & \text{Gap in } T \end{cases} \tag{6.3}$$

For filling cell $M[1,1]$, we have to calculate the maximum of $M[0,0] + \mathrm{sim}(a,b) = 0 + (-1) = -1$ (for aligning a and b) and -2 (for introducing a gap in either sequence, cf.

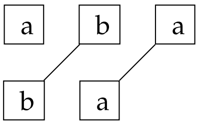


Figure 6.2: Alignment for $\langle a, b, a\rangle$ and $\langle b, a\rangle$ produced by Needleman-Wunsch

(6.3)). In this case, we align a and b and fill in $M[1,1] = -1$.

$$M_1 = \begin{bmatrix} 0 & -1 & -2 & -3 \\ -1 & -1 & & \\ -2 & & & \end{bmatrix} \tag{6.4}$$

In the next step, we fill $M[2,1]$ and choose the maximum of $M[1,0] + \mathrm{sim}(b,b) = -1+2 = 1$ (aligning b and b), $M[2,0] + (-1) = -2$ (gap) and $M[1,1] + (-1) = -3$ (gap). This time, we align b and b and fill in $M[2,1] = 1$. This way, the matrix gets filled entirely, until we reach the bottom right corner. Equation 6.5 shows the full matrix after six steps. Numbers in **boldface** indicate the chosen path.

$$M_6 = \begin{bmatrix} 0 & -\mathbf{1} & -2 & -3 \\ -1 & -1 & \mathbf{1} & 0 \\ -2 & 0 & 0 & \mathbf{3} \end{bmatrix} \tag{6.5}$$

The final alignment score can be found in the bottom right corner and is 3. The extracted alignment is shown in Figure 6.2.

6.2.2 Graph-based Predicate Clustering

The graph-based predicate clustering approach on event alignment is described in Roth and Frank (2012). As the name suggests, the algorithm uses a graph as basic data representation. Each vertex in the graph represents an event from the sequences, (weighted) edges in the graph represent similarities between the events. The graph is then clustered and events in the same cluster are aligned.

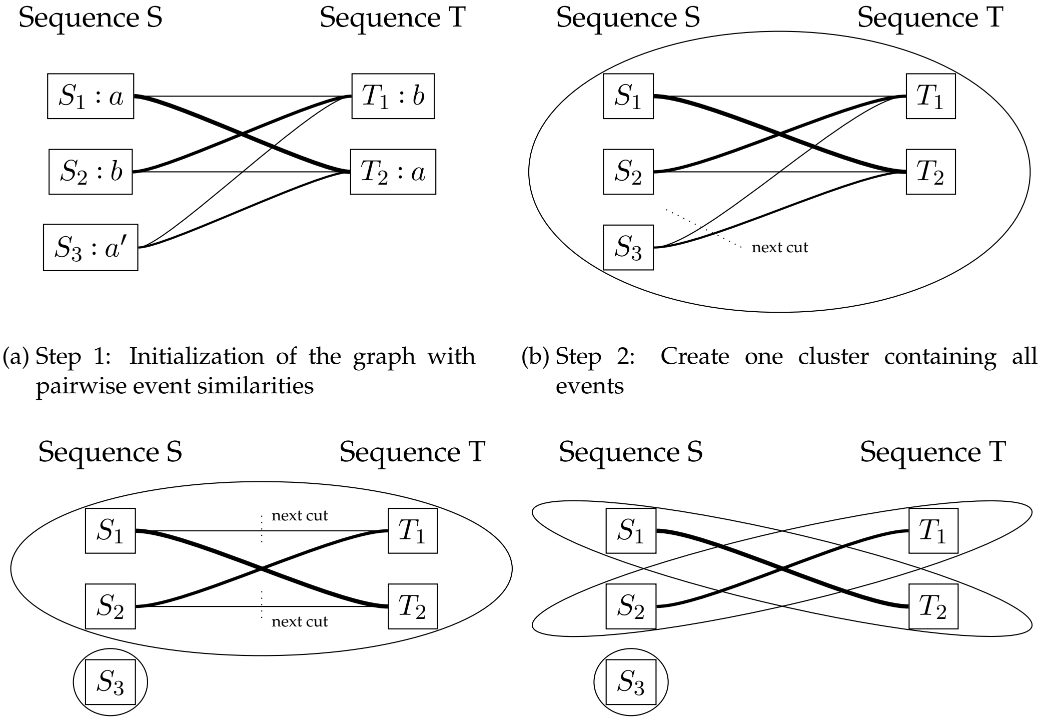
Again, we assume two sequences of events as input: $S = \langle s_1, s_2, \dots s_n\rangle$ and $T = \langle t_1, t_2, \dots t_m\rangle$. From the sequences, we construct a bipartite graph. Each event in each sequence is represented by a vertex (6.7). Edges are added between vertices iff (i) the two vertices are from different sequences and (ii) their similarity is above a lower threshold t (6.8). The similarity according to sim is attached to the edges as edge weight (6.6).

$$G = (V, E, \mathrm{sim}) \tag{6.6}$$

$$V = S \cup T \tag{6.7}$$

$$E = \{(e_1, e_2) | e_1 \in S \wedge e_2 \in T \ \wedge \mathrm{sim}(e_1, e_2) > t\} \tag{6.8}$$

In order to create alignments between events, an iterative clustering algorithm is then used to cut the graph in parts. In each iteration, the algorithm removes a number of



(a) Step 1: Initialization of the graph with pairwise event similarities

(b) Step 2: Create one cluster containing all events

(c) Step 3: Apply minimum cut

(d) Step 4: Apply minimum cut and terminate, because all clusters contain at most two events

Figure 6.3: Running predicate alignment on the sequences $\langle a, b, a \rangle$ and $\langle b, a \rangle$

edges, such that (i) the graph is cut into two unconnected parts and (ii) the summed weight of removed edges is minimal. Such a cut is called a *minimum cut* in graph theory. Roth and Frank (2012) use an implementation based on Goldberg and Tarjan (1988) to determine the minimum cut. The minimum-cut algorithm is applied iteratively, until only clusters with at most two vertices remain. The events clustered together are then extracted as an alignment.

We are using the settings and optimizations that have been optimized on newspaper texts and published in Roth and Frank (2012). A more detailed description can be found in Roth (2014).

Example

See Figure 6.3 for an example. Similarities are represented by line thickness in Figure 6.3a. We employ the same input sequences as in the previous examples. However, for the sake of the example, we assume the third event in sequence S is a slight variation (a') of the first event: $\mathrm{sim}(a, a) > \mathrm{sim}(a, a')$.

As a first step, a cluster is created that contains all events (Figure 6.3b). The first

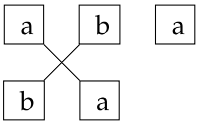


Figure 6.4: Alignment $\langle a, b, a \rangle$ and $\langle b, a \rangle$ produced by predicate alignment

cut to be applied is indicated by the dotted line (removing edges to S_3). At this point we know that S_3 will remain un-aligned. In the next step, the two edges (S_1, T_1) and (S_2, T_2) are removed, as they have minimal weight.

The algorithm terminates when all clusters contain two events or less (Figure 6.3d). In contrast to the output generated by the Needleman-Wunsch algorithm, the induced alignment contains crossing edges, as shown in Figure 6.4.

6.2.3 Bayesian Model Merging

In this algorithm, hidden Markov models (HMM) are used to represent event sequences and their overlap. Events, in HMM terminology, are observed items, while the correspondences across multiple sequences are unobserved and thus, represented by the hidden states in the HMM. Bayesian model merging (Stolcke and Omohundro, 1993) is a technique for the induction of such a HMM from sequences. The algorithm starts with the initialization of a simple HMM in which sequences have nothing in common but a start and end state. Then, hidden states are merged iteratively if the events they emit are similar.

Given a set of input sequences $\mathcal{S}$, the algorithm searches for a HMM $M \in \mathcal{M}$ that is maximally probable, given the input sequences: $\arg\max_M P(M|\mathcal{S})$. This probability can be transformed using Bayes' theorem: $P(M|\mathcal{S}) \simeq P(M)P(\mathcal{S}|M)$.

The probability of the sequences given a certain model, $P(\mathcal{S}|M)$, can easily be calculated using the forward-backward Trellis algorithm (cf. Manning and Schütze, 1999). The prior $P(M)$ needs to be defined. The general idea is to give higher probability to models with less states. In addition, the prior can be defined to yield lower probability if a state emits dissimilar events. We will first discuss how Bayesian model merging works in general and then come back to the definition of the prior.

Let $\mathcal{S} = \{S_0, S_1, \ldots, S_n\}$ be the set of input sequences over a set of events $\mathcal{E}$. In the beginning, the HMM M_0 is initialized in such a way that $\forall S_i \in \mathcal{S} : P(S_i|M_0) = \frac{1}{n}$. In words, all sequences are equally probable. Internally, each (observed) event is emitted from a hidden state and the hidden states are connected sequentially. A special start node is connected to the first hidden state of each sequence, similarly are the last hidden states of each sequence connected to a special end node.

The algorithm then works iteratively by merging two hidden states of model M_i in order to induce model M_{i+1}. In each step, the algorithm searches for a pair of states to be merged, such that $P(M_{i+1}|\mathcal{S}) > P(M_i|\mathcal{S})$. As each merge may introduce new transitions and therefore increase the number of paths through the HMM, $P(\mathcal{S}|M_i)$ monotonically decreases. This can (and should) be counterbalanced by the prior $P(M_i)$.

Finlayson (2012) used Bayesian model merging in order to automatically detect narrative structure in the form of a HMM on narrative texts that feature manually corrected linguistic annotations. We follow his general approach on defining the prior. The prior probability of a model $P(M)$ (Eq. 6.9) is a product of two functions:

$$\begin{aligned} P(M) &= \mathrm{geo}(M)\,\mathrm{plaus}(M) && (6.9)\\ \mathrm{geo} &: \mathcal{M} \to [0,1] && (6.10)\\ \mathrm{plaus} &: \mathcal{M} \to \{0,1\} && (6.11) \end{aligned}$$

As shown in Eq. 6.12, geo represents a geometric distribution that gives higher probability to smaller models ($|M|$ stands for the number of hidden states in the HMM), depending on the prior parameter $0 \le p \le 1$ (Finlayson uses $p = 0.95$). Intuitively, this makes the tendency for smaller models quite strong.

$$\begin{aligned} \mathrm{geo}(M) &= p(1-p)^{|M|-1} && (6.12)\\ \mathrm{plaus}(M) &= \prod_{\forall n \in M} K(n) && (6.13)\\ K(n) &= \begin{cases} 1 & \text{if } \forall e_i, e_j \in n, \mathrm{sim}_b(e_i, e_j) > t \\ 0 & \text{otherwise} \end{cases} && (6.14) \end{aligned}$$

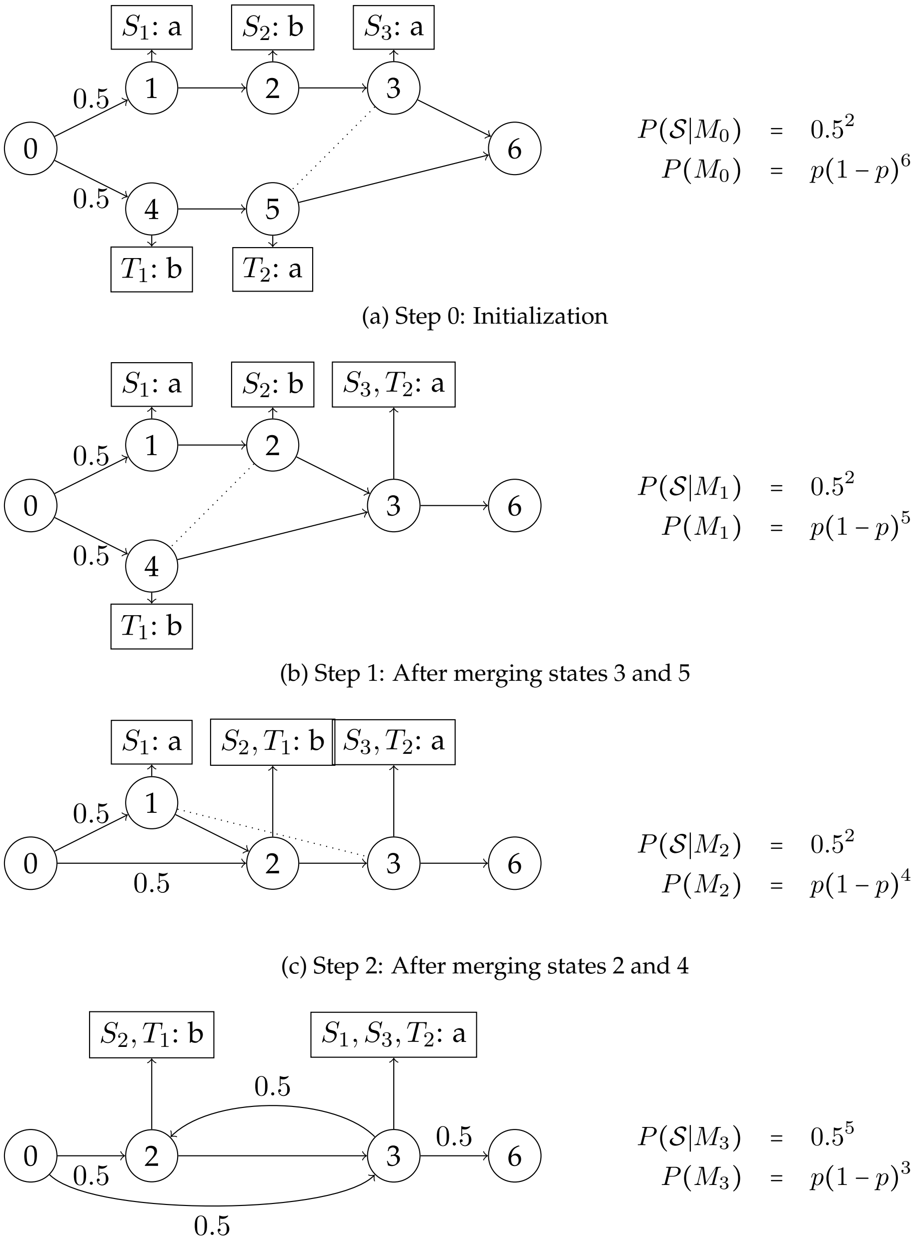
The second function $\mathrm{plaus}(M)$ (6.13) represents the 'plausibility' of the model and can only be 0 or 1. $\mathrm{plaus}(M)$ is calculated as a product over function K for all hidden states n of the model. For each state, $K(n)$ equals 1 if all pairs of events emitted from the state are more similar than threshold t. Otherwise, $K(n)$ becomes zero and so does the plausibility function for the entire model $\mathrm{plaus}(M)$. This makes the similarity threshold a hard constraint and as a result the induced alignment does not contain alignment links with a similarity lower than t.

We extract an alignment from the final HMM by creating an alignment link between all events that are emitted from the same state. This algorithm is able to create arbitrary alignment links: Crossing alignment links or links that include more than two events and documents. Also, the algorithm can generate links within a single document and thus, create cyclic structures.

Example

As an example, we use Bayesian model merging in order to induce a HMM for the set $\mathcal{S}$ of two sequences $S = \langle a, b, a\rangle$ and $T = \langle b, a\rangle$. For the sake of the example, we assume identity as similarity function, such that $\mathrm{sim}(a,a) = 1$ and $\mathrm{sim}(a,b) = 0$. We are using the prior probability as described above, the choice of the prior parameter p will only play a role in Step 3. Figure 6.5 shows each step of the application of the algorithm, starting with the initialization. In the initialization model, both sequences have equal probability:

$$P(\mathcal{S}|M_0) = 0.5^2 \qquad (6.15)$$



(a) Step 0: Initialization

(b) Step 1: After merging states 3 and 5

(c) Step 2: After merging states 2 and 4

(d) Step 3 (if $p > 0.875$): After merging states 1 and 3

Figure 6.5: Running Bayesian model merging on the sequences $\langle a, b, a\rangle$ and $\langle b, a\rangle$

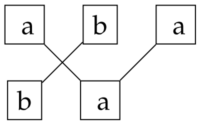


Figure 6.6: Alignment for $\langle a, b, a\rangle$ and $\langle b, a\rangle$ produced by Bayesian model merging

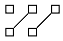
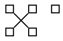
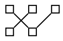
Algorithm	# events	crossing	similarity	$A(\langle a,b,a\rangle,\langle b,a\rangle)$
Needleman-Wunsch	2	–	function	
Predicate alignment	2	✓	function	
Bayesian model merging	n	✓	function	

Table 6.2: Algorithm overview

In each step, a pair of hidden states is merged. The dotted lines indicate the pair of states to be merged next. The probability of the sequences does not change in the first two steps ($P(\mathcal{S}|M_0) = P(\mathcal{S}|M_1) = P(\mathcal{S}|M_2)$), and the model probability increases ($P(M_2) > P(M_1) > P(M_0)$), because the number of hidden states decreases. Merging the states 1 and 3, however, drastically decreases the probability of the sequences. Therefore, this step will only be performed if the increase in model probability outperforms the decrease.

$$0.5^2 p(1-p)^4 < 0.5^5 p(1-p)^3 \tag{6.16}$$

$$p > 0.875 \tag{6.17}$$

In this case, (6.16) needs to be fulfilled, in order to execute step 3. In other words, p needs to be at least 0.875 in order to counterbalance the decrease in sequence probability in step 3. If this is the case, the algorithm produces two links (one of them 2-to-1) and aligns every a-event and every b-event, as shown in Figure 6.6. If not, Bayesian model merging produces the same alignment as Needleman-Wunsch.

6.2.4 Comparison of Alignment Algorithms

Table 6.2 shows an overview of three key properties of the algorithms: The number of events that can be in an alignment link, whether the algorithm can generate crossing alignment links and how similarity is measured. The Needleman-Wunsch algorithm aligns two events, but does not generate crossing alignments. In a situation in which it would be possible, it will skip sequence elements instead. The predicate alignment algorithm is able to generate crossing links, but has been developed for linking only two events. Extending it to allow n-to-m-links is possible, but has not been tested in practice[1]. Bayesian model merging is the most liberal algorithm. It aligns an arbitrary number of events and the resulting links may be crossing.

All three algorithms can be used with a similarity function that is defined externally and can be integrated in a modularized way. The algorithms also have in common that

[1] Both graph representation and clustering algorithm could be used unchanged. The exit condition, however, would need to be rethought.

they work in an unsupervised manner. No training data is needed, except for tuning the similarity weight vector parameters (see below).

6.2.5 Similarity Measures

In order to assess the similarity between two events e_i and e_j, we use several different measures of semantic similarity in combination. All of them return a value in $[0, 1]$. Apart from the first of the following measures, our implementations are based on the implementations by Michael Roth (Roth and Frank, 2012).

The measures are combined using the geometric or arithmetic mean and different weightings, as shown in (6.18) and (6.19).

$$\begin{aligned}
\mathrm{sim}_{geo}(e_1, e_2) &= \sqrt[5]{\mathrm{sim}_F(e_1, e_2)^{\lambda_F} \times \mathrm{sim}_W(e_1, e_2)^{\lambda_W} \times \mathrm{sim}_V(e_1, e_2)^{\lambda_V} \times \ldots} \\
&\quad \overline{\times \, \mathrm{sim}_D(e_1, e_2)^{\lambda_D} \times \mathrm{sim}_A(e_1, e_2)^{\lambda_A}} && (6.18) \\
\mathrm{sim}_{avg}(e_1, e_2) &= \frac{1}{5}\big(\lambda_F \, \mathrm{sim}_F(e_1, e_2) + \lambda_W \, \mathrm{sim}_W(e_1, e_2) + \lambda_V \, \mathrm{sim}_V(e_1, e_2) + \ldots \\
&\quad + \lambda_D \, \mathrm{sim}_D(e_1, e_2) + \lambda_A \, \mathrm{sim}_A(e_1, e_2)\big) && (6.19)
\end{aligned}$$

FrameNet similarity (F)

FrameNet similarity is a lexical measure and based on the FrameNet hierarchy using all FrameNet relations. If d is the length of the shortest possible path between two frames, we calculate the similarity as $\frac{1}{d+1}$. This way, frames with a distance of 0 get the maximal similarity. We are using Dijkstra's algorithm (Dijkstra, 1959) for finding the shortest path. If, for any reason, no path can be found between the two frames, the similarity is set to 0. This in particular happens for frames that are not connected in the FrameNet hierarchy.

WordNet similarity (W)

For measuring similarity according to WordNet, we are using the similarity measure introduced by Lin (1998) applied on the synsets of the frame targets, which are assigned by the word sense disambiguation component. For this measure, the information content (ic) of the lowest common subsumer of the two synsets is set in relation with the information content of the synsets itself, as shown in equation 6.20. The information content has been precomputed on the British National Corpus, the Penn treebank, the Brown corpus, the complete works of Shakespeare and SemCor (Pedersen, 2014).

$$\frac{2 * \mathrm{ic}(\mathrm{lcs}(s_1, s_2))}{\mathrm{ic}(s_1) + \mathrm{ic}(s_2)} \tag{6.20}$$

VerbNet similarity (V)

This measure detects overlap in potential VerbNet classes, calculated using the target lemmas of the frames. The resulting similarity value differentiates three cases: (i) If there is a VerbNet class that contains both verbs, the similarity is 1. (ii) If one verb is in a subclass of a class of the other verb, the similarity is 0.8. (iii) If there are only disjunct classes for the verbs, the similarity is 0.

Distance similarity (D)

This similarity measure compares the relative positions of the two events in their respective chains. First, the relative position is computed. Then, the difference of the two relative positions is calculated and normalized with a Gaussian distribution ($\sigma = 0.2$). Therefore, smaller differences in relative positioning are not penalized as much.

Argument text similarity (A)

This measure includes the arguments of the events. For both events e_i and e_j, we collect a bag of words s_i and s_j containing the lemmas of the frame element fillers. In addition, for each filler, the set of coreferent lemmas is added. The similarity is calculated according to Equation 6.21.

$$\mathrm{sim}(e_i, e_j) = \frac{|s_i \cap s_j|}{|s_i| + |s_j|} \tag{6.21}$$

6.3 Gold Standard and Evaluation

6.3.1 Data Sets

In Chapter 4, we have described two scholarly areas in which structural similarities play a major role and which we use as application scenarios. We have also described linguistic characteristics of the texts in the corpora. We will now describe how we use the corpora in order to measure the performance of the event alignment algorithms for the detection of story similarities.

Alignment Gold Standard

In order to get a detailed insight into the performance of the algorithms, we annotated a small set of descriptions of rituals manually with alignment links.

The annotated data set consists of alignment links between the cūḍākaraṇa rituals. The annotation has been performed by two experts at rituals independently and reflects the discussions in scientific literature about ritual elements. Annotating alignments across descriptions of rituals proved to be a tedious and difficult task. Although the descriptions are detailed, they are not very clear ("underspecified") in many cases. This makes it hard, even for experts, to exactly pinpoint the similarities. Researchers of

Description	# tokens	# events
A	1,986	132
C	1,071	91
I	1,162	100

Pair	# links	% 1-1 links
A, C	11	54.5 %
A, I	16	56.2 %
C, I	45	84.4 %

Table 6.3: Alignment gold standard

	#documents	#clusters	∅#tokens
Folktales	37	7	717.6
Rituals	13	5	2,040.2

Table 6.4: Overview of clustering data sets

rituals have published proposals for ritual elements, but they are not clearly defined and in particular, it is often difficult to tell where they start and end.

We did not provide an annotation interface. One annotator used a CSV file to store his annotations, the other marked them on paper. After an initial conversion of the paper-based annotations into a machine-readable file, the initial agreement between the two annotators was very low: $\kappa = 0.19$ (Fleiss' kappa; measured as a pairwise classification task). A discussion with one annotator allowed the refinement of the annotations, as he explained the comments he gave on paper. Measuring the agreement of the refined alignment results in $\kappa = 0.61$. The remaining differences have been adjudicated by the author of this thesis. The annotator approved the final alignment as a possible one.

Table 6.3 shows some statistics about the documents themselves on the lefthand side and statistics about the alignment links on the righthand side. Noteworthy on the right table is that the pair (C,I) contains many more links as any other pair. Also, most of them are one-to-one links. The alignments involving A seem to be much harder and fuzzier. This is in line with the fact that A is much longer and has a different cultural background. Multiple events in A are linked to single events in C and I and are thus 1-to-n-links.

Clustering Gold Standard

Both corpora are classified into groups according to story elements they employ: (i) The folktales are grouped into ATU classes and ATU classes are defined by shared elements in the stories. All the tales in ATU class 327A (Hansel und Gretel), for instance, have in common that children are abandoned in the woods, stumble upon a gingerbread house etc. To our knowledge, we are the first to use the ATU index to define a classification that can be used for evaluation and investigation. (ii) A subset of thirteen descriptions of rituals are grouped according to their ritual type, i.e., the ritual they describe. Similarly to the folktales, structural similarities can be expected between the descriptions of the same ritual type. Although the descriptions come from different handbooks and differ in many details, the main events should be similar and in a similar order. We will use the classifications as a gold standard in the clustering experiment. Table 6.4 lists sizes and average number of tokens per document for the two data sets.

6.3.2 Evaluation Measures

In order to quantify the performance of the algorithms, we need two evaluation measures. For comparing the performance of the alignment-based clustering with the gold clustering, we employ a classic cluster quality measure called Rand index (Rand, 1971). In contrast to purity, which assigns each cluster to its majority class and calculates accuracy per cluster, the Rand index penalizes both false positive and false negative decisions.

Finding an appropriate evaluation measure for alignment evaluation proved to be more complicated, given that it should be able to (i) cope with n-to-m-links and (ii) scale to more than two documents. The first requirement comes directly from the gold standard, which already contains a high number of n-to-m-links. The second requirement may not be so obvious, but given that one of the algorithms is capable of running on more than two documents and the gold standard also includes links across three documents, the evaluation algorithm should allow that as well.

The comparison of manually created alignments with system alignments has been researched a lot in the context of machine translation and cross-lingual word or sentence alignment. Many alignment evaluation measures break down n-to-m-links into pairwise 1-to-1-links (cf. Fraser and Marcu, 2007; Och and Ney, 2003). Tiedemann (2003) argues that this can lead to highly skewed results, in particular when n and m get large, as every n-to-m-link introduces $n * m$ pairwise links. He proposes to count every n-to-m-link as a single link, but weighted according to the correctly aligned tokens on both source and target side.

Comparing alignments can be seen as a comparison of sets. Given a set of sequences $\mathcal{S} = \{S_i | S_i = \langle s_{i,0}, s_{i,1}, \dots \rangle\}$, an alignment can be expressed as a set of alignment links and an alignment link as a set of sequence elements. Aligning, for instance, the first elements of the sequences S_0 and S_1 would then be expressed as the set $\{s_{0,0}, s_{1,0}\}$. This is very similar to coreference resolution, in which a set of sets of mentions has to be constructed and compared to a reference set of sets of mentions. Specifically for coreference resolution, the Rand-based Blanc score has been proposed as an evaluation measure. Using Blanc as a measure for alignment evaluation would allow n-to-m-links easily and it can be applied directly to alignments of multiple documents (because as a coreference resolution metric, Blanc does not know about documents at all).

Rand index

The Rand index (Rand, 1971) is a classic measure of cluster quality. It can be used to measure the quality of arbitrary partitions, in particular including partial ones. Let $X = \{X_1, X_2, \dots, X_n\}$ be the set of objects to be clustered (in our case: documents) and S and R be the system and reference partitioning. For each pair of objects (X_i, X_j), the algorithm then counts if the two objects are in the same or different cluster in S and R. Two cases can be distinguished:

a) Correct decision: X_i and X_j are in the same cluster in both S and R or in different clusters in both

b) Incorrect decision: X_i and X_j are in the same cluster in S and in different clusters in R, or vice-versa

The first case represents agreements of the system partitioning with the reference partitioning. This can mean either that X_i and X_j are in the same cluster in both system and reference partitioning or that they are in different clusters in both clusterings. The Rand index is then defined as shown in Equation 6.22, where $|a|$ is the number of a cases (the number of agreements). Simply put, the Rand index represents the portion of correct pairwise decisions.

$$\text{Rand}(S, R) = \frac{|a|}{\binom{n}{2}} \tag{6.22}$$

The Rand index is a single score, producing values between 0 (no similarity) and 1 (equal clusterings). An important property of the Rand index is that objects not in the same clusters are evaluated as well. We are using the Rand index as a measure for comparing the manually defined clusterings of tales and descriptions of rituals to the system outputs of a clustering algorithm (Experiment 2).

Blanc

Blanc (Recasens and Hovy, 2011) is an extension of the Rand index for evaluating coreference chains. Similarly to the regular Rand index, system output S and reference R are compared for each pair of mentions X_i and X_j.

a) Correct decision: X_i and X_j are in the same cluster in both S and R or in different clusters in both

b) Incorrect decision: X_i and X_j are in the same cluster in S and in different clusters in R (or vice-versa)

The cases a) and b) are counted separately for coreference and non-coreference links (two mentions are in a non-coreference link if they are not coreferent). Then, precision, recall and f-score are calculated as shown in 6.23, for coreference links ($_c$), non-coreference links ($_n$) and overall.

$$\begin{array}{lll}
P_c = \dfrac{a_c}{a_c + b_c} & P_n = \frac{a_n}{a_n + b_n} & P = \dfrac{P_c + P_n}{2} \\
R_c = \dfrac{a_c}{a_c + b_n} & R_n = \frac{a_n}{a_n + b_c} & R = \dfrac{R_c + R_n}{2} \\
F_c = \dfrac{2 P_c R_c}{P_c + R_c} & F_n = \frac{2 P_n R_n}{P_n + R_n} & F = \dfrac{F_c + F_n}{2}
\end{array} \tag{6.23}$$

An issue with the evaluation of end to end coreference resolution systems is the (possible) discrepancy between system mentions and reference mentions: A coreference resolution system may detect a different set of mentions than is annotated in the reference data set. This makes evaluation inherently difficult and has sparked a lot of debates in the coreference resolution community (see, for instance Cai and Strube (2010), for a discussion of metrics and their applicability to end to end systems). In our case, however, we can evaluate over all tokens in the documents, which are necessarily the same for system and reference.

Because coreference and non-coreference links are weighted equally, the resulting general precision, recall and f-score values are highly biased. A system that generates very few alignment links already achieves around 50% precision and recall, because the vast majority of pairs are in fact non-aligned in the gold standard. This tendency can be seen in the experiment, in which a number of configurations achieve close to 50% precision and recall.

6.4 Experiment 1: Comparison against an Alignment Gold Standard

In the first experiment, we evaluate generated alignments directly against a manually annotated gold standard. We are using the data set in order to optimize the similarity weight vector and the threshold for the Bayesian model merging and Needleman-Wunsch algorithms. We use cross validation for parameter tuning and the evaluation. All algorithms are using the targets of frames that are reliably annotated as input sequences. We select the reliably annotated frames by choosing only the frames whose targets have been annotated in the training set of the semantic role labeling component. We will first describe the experimental setup and then results including an error analysis.

6.4.1 Cross Validation

In order to test the performance of the algorithms in a reliable fashion, we optimize their parameters on two pairs and test them on the remaining pair. $\mathcal{D} = \{A, C, I\}$ is the set of documents in the gold standard and $\mathcal{C}$ the set of candidate configurations to be optimized. We run the algorithm in each configuration $c_j \in \mathcal{C}$ on each pair of documents $p_i \in \mathcal{D} \times \mathcal{D}$ and test against the gold standard. This way, two pairs will serve as optimization set and one pair as test set. The best performing configuration c'_{p_i} is extracted for each pair. We select the configuration that achieves the highest Blanc score on both optimization pairs as final configuration in order to evaluate the test pair. To be clear: This is not a training step in the classical, supervised sense, but a parameter optimization step.

6.4.2 Parameter Settings

We optimized the weight w for each similarity measure as well as the threshold and the mean calculation (geometric and arithmetic mean).

Because the Needleman-Wunsch algorithm compares different possible alignments to each other, the setting of the threshold does not make a difference: If by increasing the threshold a certain link score decreases, so do all the other link scores. Therefore, we only optimized the weight vector for the similarity measures in the above described manner. The best performing vector was equal weight for all measures and using geometric mean.

For the predicate alignment algorithm, the best performing settings $\lambda_F = 2$, $\lambda_A = 2$, $\lambda_D = 2$, $\lambda_V = 1.67$, $\lambda_W = 1$, a threshold of $t = 0.8$ and using the geometric mean for combination. The best performing weight vector for the Bayesian model merging algorithm was weighting each measure equally and using the geometric mean to combine them. The best performing threshold was $t = 0.8$ for all optimization pairs.

Over all algorithms, using geometric mean achieves better results than arithmetic mean. In addition, quite high threshold settings have been determined.

6.4.3 Baseline

We compare the results against two baselines. The *harmonic baseline* algorithm creates an alignment by linking all elements of the shorter sequence to their positional counterpart. The unaligned elements of the longer sequence are then added to the surrounding alignment links. Let, for instance, $S = \langle s_1, s_2, s_3, s_4 \rangle$ be one sequence and $T = \langle t_1, t_2 \rangle$ be the other sequence, the baseline alignment would link $\{s_1, s_2, t_1\}$ and $\{s_3, s_4, t_2\}$. Additionally, the *lemma alignment baseline* creates alignment links between all events with the same lemma. This creates many n-to-m-links. In both baseline algorithms, we use the same set of candidate events as in the other algorithms.

6.4.4 Results

Table 6.5 shows the results for all pairs of descriptions of rituals[1]. The first column displays the number of alignment links generated by the algorithm, with the number of 1-to-1-links in parentheses, if applicable (Needleman-Wunsch and predicate alignment only generate 1-to-1-links). The second to fourth columns show the Blanc scores.

In terms of precision, the predicate alignment achieves the highest score on one pair, while Bayesian model merging achieves the highest precision score on the other pairs. The highest recall for pairs AC and AI is also achieved by the Bayesian model merging, while for CI, the predicate alignment achieves a slightly higher recall.

All three algorithms show similar behavior in one respect: Performance scores on CI are higher than on the other pairs. We can explain this by looking at the nature of the documents. Compared with the other pairings, CI contains much more links and most of them are non-crossing 1-to-1-links (cf. Table 6.3), which makes it easier for the algorithms. The other two pairings, AC and AI are much harder for all algorithms. Both

		# links (#1-to-1)	Blanc-P	Blanc-R	Blanc
Lemma Baseline	A, C	14 (2)	51.1	59.5	51.4
	A, I	15 (1)	50.7	59.6	50.5
	C, I	19 (4)	50.4	57.6	49.0
Harmonic Baseline	A, C	91 (50)	50.1	50.2	50.1
	A, I	100 (68)	49.8	49.8	49.8
	C, I	91 (82)	50.3	50.3	50.3
Needleman-Wunsch	A, C	90	49.8	49.9	49.8
	A, I	97	49.8	49.9	49.9
	C, I	76	55.1	54.0	54.5
Predicate Alignment	A, C	0	49.9	50.0	49.9
	A, I	4	49.9	50.0	49.9
	C, I	24	76.9	57.0	61.0
Bayesian Model Merging	A, C	7 (7)	64.1	50.8	51.5
	A, I	10 (10)	54.9	50.5	50.8
	C, I	37 (36)	65.2	56.4	59.0

Table 6.5: Results for Experiment 1: Comparison with a gold standard

Needleman-Wunsch and the predicate alignment system perform similarly or below the harmonic baseline. In fact, they do not detect a single correct alignment between events on pairs involving A. The Bayesian model merging algorithm outperforms the harmonic baseline on every pair and the lemma baseline in terms of precision. Although the predicate alignment outperforms Bayesian model merging on the pair CI, it has to be noted that the structure Bayesian model merging induces (the hidden Markov model) is more complex and offers more insight. The alignment can be considered a "by-product" of the hidden Markov model.

Number of alignment links

The lemma baseline generates only a few links, but some of them are quite large. The largest alignment link generated by the lemma baseline contains 60 different events. This is, although some of the links are correct, not suitable for our task. The harmonic baseline as well as the Needleman-Wunsch algorithm generate close to 100 links in most cases (gold standard: AC: 11, AI: 16, CI: 45). Interestingly, the pair CI, which actually has the most links in the gold standard, is the pair that gets the fewest links assigned by the Needleman-Wunsch algorithm. This can be explained by the fact that the long ritual (A) is not involved. The predicate alignment and the Bayesian model merging

[1] This table shows the results of the different system, but separately for aligned pairs ($_c$) and non-aligned pairs ($_n$). As can be seen, the performance for alignment links is far from perfect. In comparison to each other, however, the systems performances behave similarly as in the combined scores.

generate much fewer links, and, in agreement with the gold standard, more links for CI than for the other pairs.

Error analysis

Manual inspection of the alignments generated by the systems reveals several major sources of errors.

Preprocessing errors Although we have adapted the preprocessing pipeline heavily in order to improve the linguistic annotation quality, there are errors in the pre-annotation (as can be expected). We will not discuss this issue in detail, but it is a source of errors.

Event extraction An issue is the generation of the input sequences for all the algorithms. Baseline, Needleman-Wunsch and Bayesian model merging use FrameNet frames as event representations and generate alignments between all automatically assigned frames, whose targets have been seen in the training set of the semantic role labeler. This, however, includes a number of frames that clearly do not represent events. The most prominent example is KINSHIP on targets like *mother*, *sister*, ... The straightforward answer, restricting events to be verbal, is not feasible, because many events in the descriptions are expressed as nouns (*Salutation*, *Offering*, ...). The predicate alignment system uses nominal and verbal predicate argument structures as event representations, which leads to different examples of the same problem: *boy*, *south*, ...

Arguments of frequent events A few lemmas that are of general meaning appear often in the descriptions of rituals, e.g., *place* or *take*. They often describe similar actions that only differ slightly, e.g., in the cardinal direction something should be placed.

		P_c	R_c	F_c	P_n	R_n	F_n
Lemma Baseline	A, C	2.5	22.0	4.5	99.7	97.1	98.4
	A, I	1.7	23.4	3.2	99.8	95.9	97.8
	C, I	1.0	22.6	1.9	99.7	92.6	96.0
Harmonic Baseline	A, C	0.6	0.8	0.7	99.6	99.5	99.6
	A, I	0.0	0.0	0.0	99.7	99.5	99.6
	C, I	0.9	1.0	1.0	99.7	99.6	99.6
Needleman-Wunsch	A, C	0.0	0.0	0.0	99.6	99.7	99.7
	A, I	0.0	0.0	0.0	99.7	99.7	99.7
	C, I	10.5	8.2	9.2	99.7	99.8	99.7
Predicate Alignment	A, C	0.0	0.0	0.0	99.8	100.0	99.9
	A, I	0.0	0.0	0.0	99.8	100.0	99.9
	C, I	54.2	14.0	22.2	99.7	100.0	99.8
Bayesian Model Merging	A, C	28.6	1.7	3.1	99.7	100.0	99.8
	A, I	10.0	0.9	1.7	99.7	100.0	99.8
	C, I	30.8	12.9	18.2	99.7	99.9	99.8

This information could be present as a frame element, but the argument text similarity measure does not differentiate between different frame elements. Therefore, the key difference is easily "overlooked", in particular if another measure votes for their similarity.

Reciting mantras Two often appearing constructions describe the recitation of mantras:

(12) a. Sprinkle water reciting the devasya tvā.

b. Sprinkle water with the devasya tvā.

The semantic role labeler annotates *reciting*, as in 12a, as an instance of TEXT_CREATION, with the mantra as filler of the frame element TEXT. The mantra in 12b remains unannotated or a filler of a MANNER element of a frame representing the sprinkling action. The only means of detecting the similarity of the mantras in these constructions is the similarity of argument fillers.

Knowledge bottleneck In some cases, the linguistic realizations of similar actions is so different that it would require a lot of world and/or domain knowledge to detect their similarity:

(13) a. Place cakraphaṇi on the head reciting the trātāram indram.

b. Bind a phani on the tuft reciting tava vāyav.

The events described in 13 should be aligned, according to the gold standard. Unfortunately, they differ in most aspects covered by our similarity functions: Both the appropriate concepts in FrameNet as well as WordNet are quite far of each other in the hierarchy. Except for the verb *reciting*, the arguments are different. The relative distance of the two events in their sequence is in a medium range. Therefore, they get low similarity and are not linked. A domain knowledge base might include a relation between *cakraphaṇi* and *phani*, but relating placing and binding and head and tuft requires a large amount of knowledge.

Quantified error analysis

In order to also get a quantified overview of the errors made by the systems, we classified the precision errors manually into four different classes. This classification is done on a pairwise basis, i.e., n-to-m-links are broken into 1-to-1-links. Similarly to the evaluation metric, we classify each pair of events. We grouped the errors into three different classes plus a rest class: (i) Events that have different arguments and should not be aligned, (ii) events that have the same arguments and should not be aligned, (iii) Events that are no events, i.e., preselection errors and (iv) Other errors.

Table 6.6 shows the portions of errors made by the predicate alignment and Bayesian model merging system on the pair CI. What we can see here is that the Bayesian model

	i	ii	iii	iv
Predicate alignment	45.5%	45.5%	0.0%	9.1%
Bayesian model merging	65.5%	19.2%	7.7%	7.7%

Table 6.6: Quantified error analysis of precision errors in Experiment 1

merging and the predicate alignment system make different (precision) errors. Most of the errors made by the predicate alignment system are caused by incorrectly linking events with different arguments or similar arguments. Two thirds of the errors made by the Bayesian model merging are events that have different arguments. This reflects the different weights the argument text similarity measure has: In the predicate alignment system, the argument similarity is assigned a high weight, while the measures are all equally weighted in the Bayesian model merging system. This could be easily changed, but the optimization showed that the overall results decreases with other weighting schemes.

6.5 Experiment 2: Alignment-based Clustering Evaluation

We use the induced alignments as an indicator of document similarity in the second experiment. We build on the fact that both the descriptions of rituals from the core corpus as well as the folktales are grouped according to their event structure: Rituals are grouped according to their ritual type, tales are grouped according to the ATU index. The intuition is that two documents from the same group share more alignment links than two documents from different groups. Or, the other way around: If an algorithm introduces many alignments across documents, they should belong to the same group. This way, we can induce a clustering of the documents, based on the automatically assigned alignments. The induced clustering can be compared to the gold clustering as given in the corpora.

6.5.1 Document Similarity

The clustering builds on a measure $\mathrm{sim}_{\mathrm{doc}} : \mathcal{S} \times \mathcal{S} \times \mathcal{A} \to [0,1]$ for pairwise similarity of documents. This measure is defined so that it can be calculated from any cross-document alignment, independent from the algorithm that created it. Let $S \in \mathcal{S}$ and $T \in \mathcal{S}$ be the two sequences that are aligned and A the set of alignment links generated by the algorithm. We first compute the similarity within each alignment link $a \in A$ as the average similarity of all cross-document pairs (6.24).

$$\mathrm{sim}'(a) \quad = \quad \frac{\sum_{s_i \in S, t_j \in T} \mathrm{sim}(s_i, t_j)}{|\{(s,t) | s \in S \wedge t \in T\}|} \tag{6.24}$$

$\mathrm{sim}_{\mathrm{doc}}$ is then calculated as the sum of the similarity scores of all alignment links, divided by the length of the shorter sequence (6.25).

$$\mathrm{sim}_{\mathrm{doc}}(S,T,A) \quad = \quad \frac{\sum_{a \in A} \mathrm{sim}'(a)}{\min(|S|,|T|)} \tag{6.25}$$

This definition ensures that densely aligned sequences are considered to be similar. It also ensures that the document similarity is more than zero, as long as there is a single alignment link with non-zero similarity.

6.5.2 Clustering Algorithm

We employ the group-average agglomerative hierarchical clustering method (Manning and Schütze, 1999). First, the document pairs are ranked according to $\mathrm{sim}_{\mathrm{doc}}$ and each document is placed in its own cluster. Then, in each turn, the most similar clusters are merged. Cluster similarity is measured by the average document similarity. The algorithm runs until all clusters are merged. This gives us a number of different partitions as a result, all with different numbers of clusters k. We will look at two ways of selecting k: (i) The partition with the correct number of clusters (which is "oracle" information, as we only know that from the gold standard) and (ii) the partition with the maximal variance ratio. The variance ratio criterion (Caliński and Harabasz, 1974) has been proposed as a means for selecting a partition in a clustering scenario and balances between similarity within and across clusters. We will refer to these two variants as the VRC and NUM variant in the following.

A word on naming: In the following we will use the names of the alignment algorithms to refer to the clustering algorithm that uses the alignments produced by the alignment algorithm. For instance, by referring to the Bayesian model merging algorithm within the context of this experiment, we actually refer to the clustering algorithm that uses document similarity calculated on the basis of the alignments produced by the Bayesian model merging algorithm.

6.5.3 Baseline

We employ three baselines. *Lemma baseline* uses the alignment baseline from the first experiment in order to generate alignments, from which we calculate $\mathrm{sim}_{\mathrm{doc}}$ as described above, assuming all similarities to be 1. This is the only baseline that is based on actual alignment links. The *lexical overlap* baseline calculates document similarity directly over all lemmas of the documents (without inducing an alignment first). If L_0 and L_1 are the two sets of lemmas of documents D_0 and D_1, the document similarity is calculated as shown in (6.26).

$$\mathrm{sim}_{\mathrm{doc}}(D_0,D_1) = \frac{2 * |L_0 \cap L_1|}{|L_0| + |L_1|} \tag{6.26}$$

In order to also compare to shallow semantic similarity measuring approaches, we also use a *vector similarity* baseline, as implemented in the Semilar toolkit (Rus et al., 2013). We did not do any domain adaptation, but used the internal preprocessing components

Cluster criterion		NUM: $k = 5$	VRC: $k = \arg\max_k vrc(k)$			
	⌀#links	Rand	k	⌀size	σ	Rand
Gold			5	2.6	0.5	
Lemma alignment	31.3	50.0	7	1.9	1.5	70.5
Needleman-Wunsch	68.1	69.2	4	3.3	1.7	69.2
Predicate alignment	9.9	66.7	11	1.2	0.4	83.3
Bayesian model merging	13.6	69.2	7	1.9	1.5	75.6
Vector similarity		61.5	6	2.2	2.2	82.1$\star$
Lexical overlap		64.1	5	2.6	2.5	64.1$\star$

Table 6.7: Results for Experiment 2: Cluster induction on descriptions of rituals

as they are integrated. The baseline generates word vectors for each document and computes the dot product between the two.

6.5.4 Results

Table 6.7 shows the results of the clustering experiment for the descriptions of rituals, Table 6.8 shows the results for the folktales in the same way. The tables show results for both ways of choosing the number of clusters k. The columns displayed are the average number of alignment links produced between a pair of stories (⌀#links, if applicable), the Rand score for using the "oracle" k, k when using the variance ratio criterion, the average size of the clusters and the standard deviation for the sizes (⌀size, σ). Further, the tables display the Rand index score for the VRC variant and whether the difference to the next lower performing partition is statistically significant (using a t-test with $\alpha = 0.05$).

Number of links The number of links produced between two documents are generally smaller for the folktales than for the rituals, which can easily be explained by the fact that the folktales are shorter. Comparing the systems, we can observe a similar behavior for both scenarios: Needleman-Wunsch generates many links, followed by the lemma alignment baseline and the predicate alignment. Bayesian model merging generates the fewest links.

Overview The best performing algorithm is predicate alignment on both data sets. On the descriptions of rituals, it achieves a (VRC-)score of 83.3 on descriptions of rituals and 82.4 on folktales. In the NUM variant, the scores are much closer on descriptions of rituals than on folktales. Lemma alignment baseline performance on folktales is surprisingly high: With a Rand score of 83.2 it outperforms all other alignment-based algorithms.

We will first look at the rituals-scenario in detail and then at folktales.

Descriptions of rituals

From the fact that several of the algorithms achieve reasonable results, we first of all can conclude that a clustering based on structural similarities is in principle able to replicate the ritual types. This supports the initial hypothesis, that rituals of a given type indeed share structural similarities that can be represented in terms of alignments.

Generally, the results are in line with the results from Experiment 1. Predicate Alignment and Bayesian model merging achieve good results in the first experiment and are ranked first and second in the clustering experiment. The Needleman-Wunsch algorithm and the lemma alignment baseline performed poorly in the first experiment and achieve low scores in the clustering experiment. At least for the descriptions of rituals, with a carefully defined classification of the descriptions, the clustering performance seems to be indicative for the quality of the individual alignments.

The fact that the lemma alignment baseline achieves relatively low scores (in both variants) indicates that the event structure plays an important role in determining similarity of narrative structure in rituals.

Choosing k There is only one setting in which the correct number of clusters (five) is selected by the variance ratio criterion: The lexical overlap baseline. Needleman-Wunsch tends to generate fewer, but larger clusters. The clusters produced by Needleman-Wunsch are relatively homogeneous regarding their size ($\sigma = 1.7$).

Predicate Alignment induces eleven clusters on descriptions of rituals, which is only slightly below the number of descriptions (thirteen). In fact, two clusters have been created that contain more than one description, all others just consist of a single description. This is caused by the very low number of alignment links induced between the descriptions. On average, 9.9 alignment links are created between two descriptions. This causes document similarity values to be very low. However, the clusters that have been created are correct.

Bayesian model merging and the lemma alignment baseline induce more smaller clusters, that are also even more homogeneous regarding their size ($\sigma = 1.5$ in both cases). Both tend to make finer distinctions between types than Needleman-Wunsch (at a slightly better quality level). This could indicate that a finer distinction of types of rituals based on event alignments might be feasible and justified by the event structure. However, only manual inspection by a domain expert could confirm that.

Choosing k based on the document similarities computed from the algorithms (VRC) generally improves the results compared to a fixed setting (NUM). This can be expected.

Baselines In the NUM setting, the performance of the shallow baselines (vector similarity and lexical overlap) is on par with the alignment algorithms. Using the variance ratio, however, benefits the vector similarity substantially. This indicates that it is possible to achieve better performance in measuring pure document similarity. Obviously, measuring document similarity in this way does not help in locating the exact similarities and allows only very limited insight into structural similarities.

Cluster criterion		NUM: k = 7	VRC: $k = \arg\max_k vrc(k)$			
	∅#links	Rand	k	∅size	σ	Rand
Gold			7	5.3	1.5	
Lemma alignment	7.4	38.4	16	2.3	1.4	83.2⋆
Needleman-Wunsch	12.6	75.9	6	6.1	4.4	75.7
Predicate Alignment	1.9	83.5	6	6.2	3.4	82.4
Bayesian model merging	0.5	38.4	12	3.1	3.8	74.9⋆
Vector similarity		87.7	8	4.6	2.6	96.1⋆
Lexical overlap		53.0	5	7.4	11.1	49.1

Table 6.8: Results for Experiment 2: Cluster induction on folktales

Folktales

Findings for the folktale corpus are somewhat different. In the NUM variant, the performance of Bayesian model merging ranges below the baselines and all other algorithms. This is likely due to the fact that the tales are shorter than descriptions of rituals and Bayesian model merging favors precision over recall. This leads to very few alignment links between the tales (0.5 on average) and makes it hard for the algorithm to find good clusters to be merged. In addition, its threshold and similarity measures have been tuned on descriptions of rituals.

In the VRC variant, the algorithm makes fewer merges (produces more clusters than in the gold standard) and achieves higher performance (74.9). This shows that the algorithm works well on fables in principle, and that its alignments induce appropriate clusters as long as it is not forced to make merges.

The predicate alignment algorithm is the best performing algorithm on folktales as well, in both variants (VRC: 82.4, NUM: 83.5). Similarly to Bayesian model merging it is reluctant to create alignment links (1.9 on average), although to a lesser degree. In contrast to the experiment on descriptions of rituals, predicate alignment is slightly outperformed by two baselines on folktales.

The lemma alignment baseline achieves similar performance as Bayesian model merging in the NUM variant. If, however, we use the VRC variant on the lemma baseline, the performance is much higher (83.2) yet with a higher number of clusters (sixteen). This could indicate that this overall more topical (rather than structural) way of modeling similarity makes finer distinctions (sub classes) within the ATU classes that may or may not be related to structural properties.

Choosing k The behavior of the algorithms with respect to k is generally similar to the setting using descriptions of rituals. Bayesian model merging and the lemma alignment baseline induce much more clusters than in the gold standard when using the VRC variant. Again, the clusters induced seem to be more fine-grained than the ones present in the gold standard.

Looking at the other direction, the lexical overlap baseline generates fewer clusters than in the gold standard, both with a very high standard deviation ($\sigma = 11.1$). It creates a single large cluster and four smaller ones.

The predicate alignment algorithm induces six clusters in the VRC variant. This is close to the gold standard number. On descriptions of rituals, predicate alignment induced much more clusters than in the gold standard, caused by a very low number of alignment links created. Although the absolute average number of alignment links on folktales is even lower, the folktales are generally much shorter. In the case of folktales, the predicate alignment seems to have created alignment links well suited for clustering purposes.

Summary

The experiments show that event alignments can be used to induce clusters with a good overlap with gold classes in most cases (predicate alignment outperforms specialized shallow baselines on descriptions of rituals, and almost on par with baseline performance on folktales). Very sparse event alignment links, however, are fatal (Bayesian model merging on tales). High baseline performance on tales could indicate that topical similarity might play a role in ATU classes (in contrast to narrative structural similarity). Issues with the event preselection, as discovered in the first experiment, also play a role here.

6.6 Graph-based Detection of Structural Similarities

Having generated a large amount of alignments across narratives is an important step towards the discovery of structural similarities. As a means to detect structural similarities on a large scale, we have developed a graph-based algorithm that identifies events which are placed in structurally similar regions. The algorithm works on the alignments produced by any event alignment algorithm.

Assuming two sequences S, T and an alignment A. The first step is the conversion of the alignment data structure in an undirected graph $G = (V, E)$ in which events are represented as vertices (6.27). Two events are connected with an (unweighted, undirected) edge in two cases (6.28): (i) If the two events are from the same document, they are connected if one directly succeeds the other in the narrative ($v_1 \rightarrow v_2$). (ii) If two events are from different documents, they are connected if an alignment link has been produced between them. n-to-m-links are broken down into pairwise links. This creates an undirected graph as shown in Figure 6.7. The node set $\{a_1, a_2, b_1, b_3\}$ would be a structurally similar region that we seek to identify.

$$\begin{aligned} V &= S \cup T && (6.27) \\ E &= \{(v_1, v_2) | v_1 \rightarrow v_2 \vee \{v_1, v_2\} \in A\} && (6.28) \end{aligned}$$

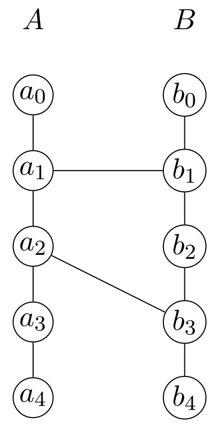


Figure 6.7: Undirected graph g created from an alignment

Our algorithm works iteratively and assigns each vertex a numerical score $\mathrm{c} : S \times T \times A \to [0, k]$ that represents its connectivity to the other sequence. This is done by starting a random walk (cf. Bollobás, 1998) of k steps, once from each vertex. The random walk selects the next visited vertex at random (equal probabilities for all possible vertices) and generates an ordered set of k vertices (6.29).

$$\mathrm{rwalk} : G \times V \times k \to V^k \tag{6.29}$$

If a vertex v has a degree of $\deg(v) = 0$, we define the random walk to be $\mathrm{rwalk}(G, v, k) = \langle v \rangle_0^k$. In this case, we basically assume a looping edge connecting v with itself. This can only happen if an input sequence is of length 1 and the one event is not aligned to any other event and is therefore not happening in practice.

We count the number of times we cross from one event sequence to another during the random walk. After one iteration (doing one walk starting in each vertex), each vertex has a score between 0 and k, which is the absolute frequency of steps that have crossed to another sequence (6.30). This is repeated n times and the scores for each vertex are added.

$$\mathrm{c}(g, v, k) \quad = \quad |\{(v_1, v_2) | v_1 \to v_2 \in \mathrm{rwalk}(g, v, k) \wedge v_1 \in S \wedge v_2 \in T\}| \tag{6.30}$$

As an example, we consider a few random walks of length $k = 2$ in Figure 6.7. If the walk would be $\mathrm{rwalk}(g, a_0, 2) = \langle a_0, a_1, a_2 \rangle$, vertex a_0 gets a score of $c = 0$, because no crossing to the other sequence has occurred. If the walk is $\langle a_0, a_1, b_1 \rangle$, a_0 gets a score of $c = 1$. The walk $\langle a_1, b_1, a_1 \rangle$ would get a score of $c = 2$.

The relative frequency of crossing the sequences when starting from a given node can easily be calculated by dividing the absolute frequency in each node by $n * k$. After two iterations ($n = 2, k = 3$) with the two walks $\langle a_0, a_1, a_2 \rangle$ and $\langle a_0, a_1, b_1 \rangle$, a_0 gets a score of $c = 1$. The relative frequency of crossing sequences is then $\frac{1}{6}$, because one of six steps

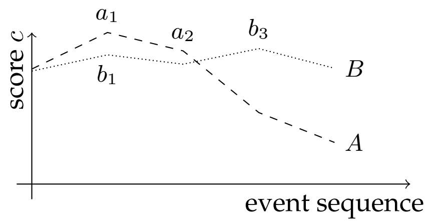


Figure 6.8: Visualisation of connectivity scores for alignment in Figure 6.7

has crossed the sequences. By increasing n, this relative frequency converges on the probability of crossing to the other sequence at least once.

Figure 6.8 shows, as an example, how the connectivity scores can be visualized for the example alignment from Figure 6.7. The scores have been calculated with $k = 3$ and $n = 1,000$. Each line represents an event sequence. The y-axis shows the connectivity scores, the x-axis the sequence ordering. In this case, the dashed line represents the left event sequence from Figure 6.7. As one would expect, the connectivity scores decrease towards the end of the sequence, because the last two event nodes are unconnected (starting from the last node a_4, there is only a single walk of length $k = 3$ that would cross to the other sequence). It is also noteworthy that the top scores in sequence A (a_1, a_2) are higher than the top scores in sequence B (b_1, b_3). This nicely represents the fact that the alignment links in A are more dense than in B.

The top ranked events according to this score can easily be extracted and represent the most connected events across two narratives. Due to being based on event alignments, "best connected" events are the most similar events both individually and structurally.

6.7 Summary

In this chapter, we have described and evaluated the technical methodology for discovering structural similarities across narrative texts. More fundamentally, we have described three different alignment algorithms that can be employed for the alignment of events. We have evaluated their performance in two experiments with mixed results. Finally, we have described a graph-based algorithm that detects dense regions of alignments across documents.

7 Analyzing and Exploiting Structural Similarities in Digital Humanities

In this chapter, we will describe in a showcase scenario for the analysis of descriptions of rituals how results from previously discussed algorithms and methods can be (i) visualized and (ii) put to use by researchers from digital humanities. In Section 7.1, we will focus on story similarities from a global perspective, comparing entire stories. Section 7.2 shows how to identify densely connected regions within pairs of descriptions and what kind of insights can be drawn from these. In Section 7.3, we focus on a specific region that can be found in this way among the descriptions of rituals.

7.1 Inspecting Story Similarities Globally

Suppose we are working in a large-scale research scenario and we have induced similarities (based on event alignments as established in Chapter 6) for a large number of documents. A first overview of the generated similarities can be gained by looking at heat maps. Figure 7.1 shows a heat map that displays the similarities between descriptions of rituals based on the Bayesian model merging. The darker a small rectangle is, the more similar the two documents are. Obviously, the diagonal rectangles are all black, because each document is maximally similar to itself. The larger rectangles represent the predefined ritual types. Ideally, the small rectangles within a large rectangle would be dark, and the small rectangles outside a large rectangle bright. In order to improve visibility, the similarity scores have been scaled.

Heat maps like these can serve as an entry point for a detailed analysis by the researcher of rituals. What we can directly see in Figure 7.1 is a dark group of rectangles surrounding the box of anna-prāśana rituals (t4), consisting of the descriptions E, F, G and H. The fact that the descriptions E and H (within the (t4)-box) are relatively similar is in line with the gold standard, as both rituals are anna-prāśana rituals. The descriptions F and G are also measured as similar, in contrast to the gold standard. F belongs in the (t5/mekhalā-bandhana)-group, while G is a nāmakaraṇa-ritual (t3). In this case, the researcher of rituals is able to inspect the alignments found between F and G and can either discover unexpected similarities or errors in the processing or clustering.

Another interesting group consists of the descriptions A, B, L and N, because their columns and rows are relatively bright in general and even seem white at some points (e.g. L with A, I, B, D, H and K)[1]. This is caused by low similarities to other descriptions of rituals. Again, an inspection of the textual sources reveals that this can indeed be

[1] The actual similarity values are slightly above zero.

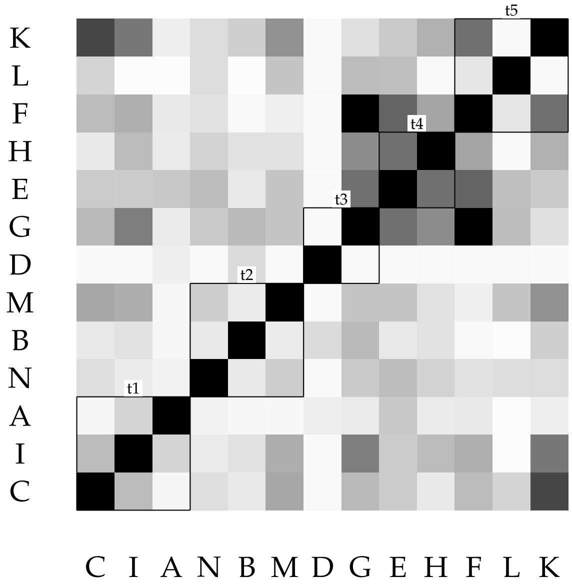


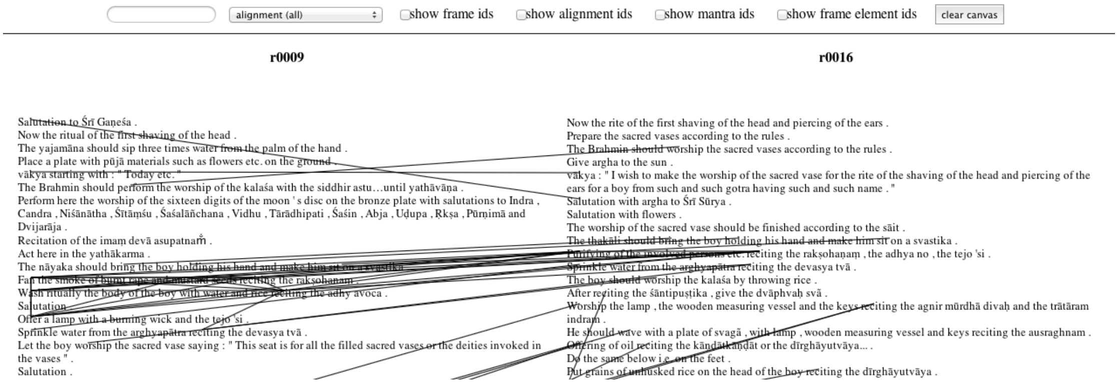
Figure 7.1: Heat map with document similarities for descriptions of rituals, based on the Bayesian model merging

explained: B is a descriptive text and therefore different from the other, prescriptive texts. A, L and N have a different cultural background and, in terms of the ritual actions, feature different elements at the beginning and ending sections.

7.2 Uncovering Structural Similarities

Heat maps provide ways for an abstract and global inspection of similarity scores. In order to analyze details, in particular alignments between specific documents, a straightforward visualization is shown in Figure 7.2. In this web-based view, the text documents are shown in parallel next to each other. Alignments are displayed as lines between frame targets and can be manually inspected. The results of the linguistic analysis can be inspected with tool-tips, the display of various metadata information can be toggled on or off. In the screenshot in Figure 7.2, we see a dense section of alignments, found automatically on the descriptions C and I using Bayesian model merging. If the full descriptions are shown, this view allows the direct visual identification of dense areas which feature similar actions in parallel. Given that we are interested in similar elements across texts, these dense areas are worth closer inspection.

In large-scale studies, the manual, visual identification of interesting areas is no longer feasible. In order to preselect interesting areas automatically, we employ the graph-based random-walk algorithm described in Section 6.6 in order to identify strongly connected components across multiple sequences. The following analysis is based on the alignments created by the Bayesian model merging algorithm. We used the algorithm with $k = 5$ and $n = 1,000$ (random walks of 5 steps length and repeating one



r0009

Salutation to Śrī Gaṇeśa .
Now the ritual of the first shaving of the head .
The yajamāna should sip three times water from the palm of the hand .
Place a plate with pūjā materials such as flowers etc. on the ground .
vākya starting with : " Today etc. "
The Brahmin should perform the worship of the kalaśa with the siddhir astu…until yathāvāṇa .
Perform here the worship of the sixteen digits of the moon ' s disc on the bronze plate with salutations to Indra , Candra , Niśānātha , Śītāṃśu , Śaśalāñchana , Vidhu , Tārādhipati , Śaśin , Abja , Uḍupa , Ṛkṣa , Pūrṇimā and Dvijarāja .
Recitation of the imaṃ devā asupatnaṁ̊ .
Act here in the yathākarma .
The nāyaka should bring the boy holding his hand and make him sit on a svastika .
Fan the smoke of burnt rape and mustard seeds reciting the rakṣohanam .
Wash ritually the body of the boy with water and rice reciting the adhy avoca .
Salutation .
Offer a lamp with a burning wick and the tejo 'si .
Sprinkle water from the arghyapātra reciting the devasya tvā .
Let the boy worship the sacred vase saying : " This seat is for all the filled sacred vases or the deities invoked in the vases " .
Salutation .

r0016

Now the rite of the first shaving of the head and piercing of the ears .
Prepare the sacred vases according to the rules .
The Brahmin should worship the sacred vases according to the rules .
Give argha to the sun .
vākya : " I wish to make the worship of the sacred vase for the rite of the shaving of the head and piercing of the ears for a boy from such and such gotra having such and such name . "
Salutation with argha to Śrī Sūrya .
Salutation with flowers .
The worship of the sacred vase should be finished according to the sāit .
The thakāli should bring the boy holding his hand and make him sit on a svastika .
Purifying of the involved persons etc. reciting the rakṣohaṇaṃ , the adhya no , the tejo 'si .
Sprinkle water from the arghyapātra reciting the devasya tvā .
The boy should worship the kalaśa by throwing rice .
After reciting the śāntipuṣṭika , give the dvāphvaḥ svā .
Worship the lamp , the wooden measuring vessel and the keys reciting the agnir mūrdhā divaḥ and the trātāram indram .
He should wave with a plate of svagā , with lamp , wooden measuring vessel and keys reciting the ausraghnam .
Offering of oil reciting the kāndātkāṇḍāt or the dīrghāyutvāya... .
Do the same below i.e. on the feet .
Put grains of unhusked rice on the head of the boy reciting the dīrghāyutvāya .

Figure 7.2: Screenshot of the alignment visualization

type	pair	15 top ranked events (sorted according to ranking)
cūḍākaraṇa (first hair cut)	AC	give(razor,barber) give(barber,razor) shave(barber,rest) throw(rice,everybody) throw(hair) sit(boy) say(forehead) shave(head) place(fruit) say(karavāṇi,patron) recite(mano jūtir,priest) place(water) keep(barber,śikhā) place(boy) sit(boy)
	AI	father(father) give(mother,portion) shave(barber,head) give(barber,piece) father(father) father(father) touch(with,hair) father(father) shave(barber,rest) recite(he,it) collect(hair) father(father,sister) take(razor) mother(mother) touch(who,need)
	CI	recite(gandhadvārāṃ) sprinkle(arghyapātra,recite) recite(trātāram indram) recite(tejo 'si) sit(boy) recite(yāḥ phalinīr) sit(boy) recite(devasya tvā) recite(ya bhūriścarā divaṃ) throw(boy,rice) recite(rakṣohanaṃ) place(cakraphaṇi) recite(ausraghnam) place(sesame) shave(hair)
anna-prāśana (first food)	EH	recite(asuraghnam) place(place) recite(hiraṇyavarṇāṃ) place(ornament) recite(yāḥ phalinīr) take(fire) recite(svastivācana,Brahmin) put(grain) offer(pañcabali) put(rice) recite(yāḥ phalinīr) take(thāybhū) offer(leaf) recite(svastivācana) put(coconut)

Table 7.1: Most connected events across descriptions A, C and I

thousand times).

First, the algorithm generates a ranking of the events according to their connectivity score c. Table 7.1 shows the top ranked 15 events for four different pairings of rituals. *Reciting*-events and non-events are colored gray. Pronouns have been replaced by the noun they refer to in order to increase readability. The top three pairs show cūḍākaraṇa rituals, which are about the first shaving, the bottom pair is an anna-prāśana ritual, which is about the first solid food that is fed to a child.

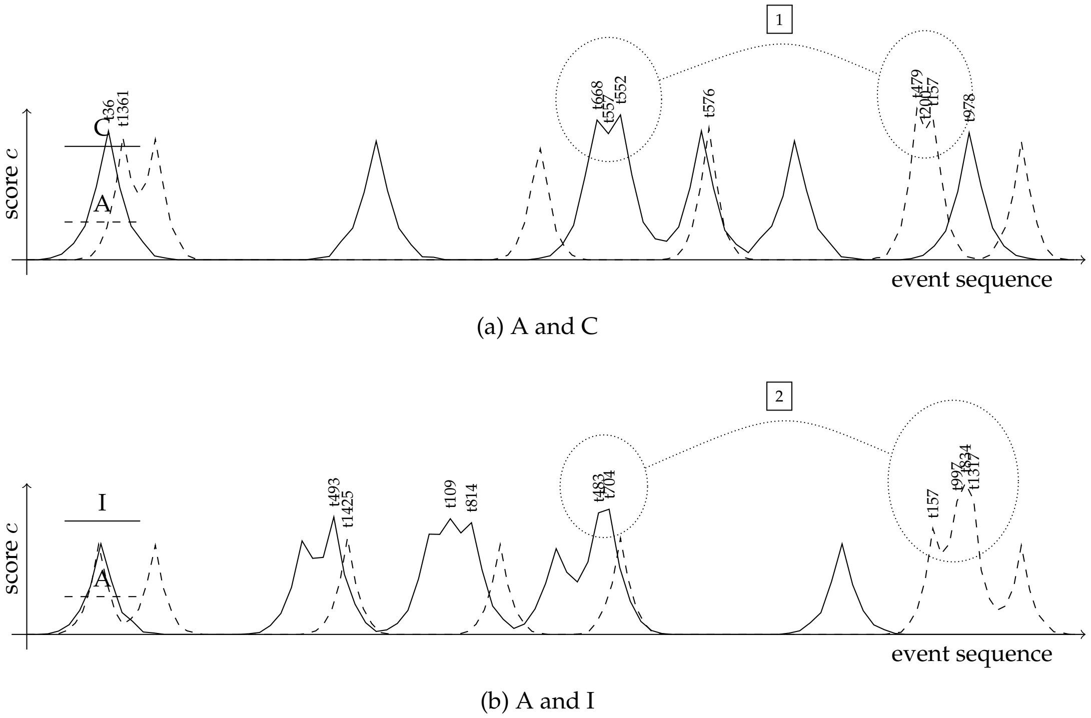
This is reflected in the list of important events: The action *shaving* appears in the first three pairs, across AI even multiple times. In terms of characters (as they are apparent in the arguments), the barber and the boy apparently play an important role. This is in line with the expectations and shows how the alignment algorithms can be put to use in order to extract important ritual elements. For the description pair EH, this is not so obvious. Although the feeding of the child appears in both descriptions, this event is not among the top 15. However, the food is still represented prominently among the event arguments: *rice*, *grain*, *coconut*, *leaf*.

Looking at important events across the ritual types, we find that rice also plays an important role across CI, but in a different kind of event (it is thrown). The same mantra also appears across the ritual types, *yāḥ phalinīır*. Although there are some similarities across types among the top ranked events, the differences are striking and in fact reflect the ritual type.

In Figures 7.3 and 7.4 we show the connectivity scores generated from three pairwise alignments of these descriptions in a graphical form (cf. Section 6.6 and Figure 6.8). The top ranked 5 events of each sequence are marked with their token id. Clearly, the figures directly represent the fact that the description pair CI is more similar than the other pairs, by showing generally higher connectivity scores in Figure 7.4 than in Figure 7.3. This is expected from analyzing the gold standard in the context of the alignment experiment.

Both pairs involving description A (Figures 7.3a and b, the dashed line shows A) show a peak close to the end of A. The alignment links that produce this peak point to a certain region in the other description, as indicated by the dotted ellipses and links marked with $\boxed{1}$ and $\boxed{2}$. In both pairs, the peak in A involves the (same) token `t157` which represents a *shaving* event. The fact that the same region from A is highly connected to two other descriptions highlights the importance of the region for the specific ritual type. Also, if we are looking at the actual context of the shaving-event in the source documents, we find other similarities (that the alignment algorithm did not capture): Before the barber shaves the hair, the razor is given to him. After the shaving, the hair is thrown into the water. This is in fact a structural similarity that goes beyond individual alignment links.

The connectivity scores shown in Figure 7.4 are generally much higher, as can be expected. Across the two descriptions C and I, we find two densely connected regions, indicated with $\boxed{3}$ and $\boxed{4}$. First of all, the fact that the most densely connected regions across the descriptions of these two rituals are at the beginning and end of the sequences can be explained by the fact that both have a similar cultural background



(a) A and C

(b) A and I

Figure 7.3: Structural similarities across pairs involving description of ritual A

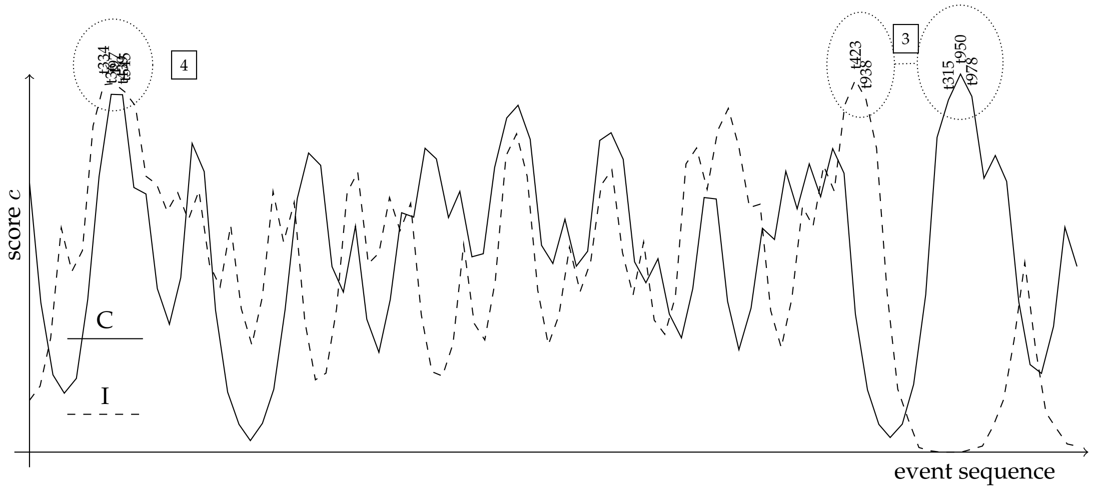


Figure 7.4: Structural similarities across CI

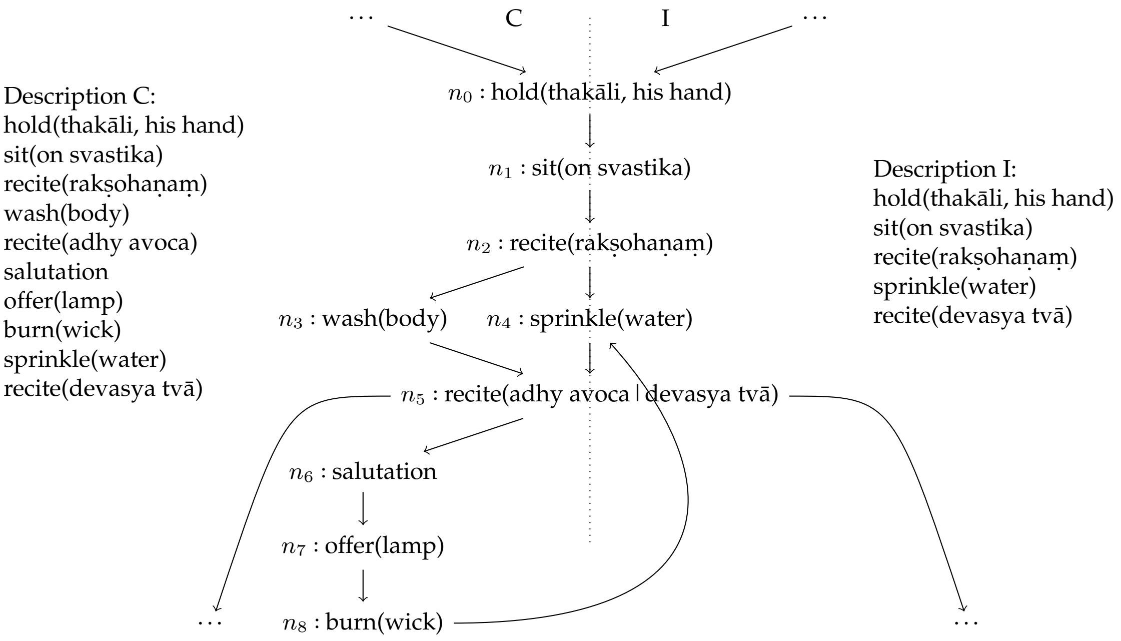


Figure 7.5: Region $\boxed{4}$ from Figure 7.4

and share most of the beginning and end. The regions marked with $\boxed{3}$ indicate structurally similar event (sub) sequences. Many of the individual events are similar and they jointly populate a dense region, indicating high structural similarity.

The region marked with $\boxed{4}$ seems like a dense heap of events. We will therefore analyze this region in the next section more closely.

7.3 Fine-grained Analysis of Structural Similarities

For the closer inspection of $\boxed{4}$, we can delve even deeper and look at the individual events that are described in both sequences. Figure 7.5 shows the densely connected region, including non-aligned events. Each event is represented by a node, aligned events have been merged into a single node, as is (conceptually) done in the Bayesian model merging algorithm. The lefthand part of the figure contains events from description C and the righthand part from description I. The events on the dotted middle line are from merged hidden states, i.e., have been aligned. The node sequences, as they appear in the texts, are: $\langle n_0, n_1, n_2, n_3, n_5, n_6, n_7, n_8, n_4, n_5\rangle$ for C and $\langle n_0, n_1, n_2, n_4, n_5\rangle$ for I and are printed as readable predicate argument structures on the far left and right.

In this region, the descriptions differ in their granularity. The first three events, represented by nodes n_0, n_1 and n_2 are completely parallel. After that, I contains a sprinkle-event (n_4) and a recitation of a mantra (devasya tvā, n_5) and then goes on. In C, however, a number of events (wash, recite, salute, offer, burn) happen before the sprinkling and the recitation of the same mantra. In C, actions are mentioned and in part described in more detail that do not appear in I. The fact that the sequence for C contains the same node twice (n_5) is actually an error. Two mantras are recited in C, one between washing and salutation and the other one after n_8, burning and n_4, sprinkling. Unfortunately, these two recitations have been merged by the algorithm.

Despite errors made by the alignment algorithm, we have detected a dense region that can be considered as a prime candidate for a ritual element during the preparation phase in Newar rituals. It is described in different granularity in the two texts. From our point of view, there are two possible reasons for this difference: Imprecision or underspecification in the writing of the description or differences in the actual execution of the ritual. This question, however, needs to be traced down and interpreted by researchers of rituals.

8 Conclusions

In this thesis, we have described a methodology for the discovery of structural similarities across narrative texts and its implementation. The system makes use of event alignment algorithms that work on linguistically analyzed texts. A full-fledged linguistic discourse analysis is done fully automatically, taking domain adaptation issues into account. The automatic discovery of similarities across narratives opens a path to scalable, empirical research in many humanities areas.

8.1 Challenges for Computational Linguistics

We will briefly discuss how the challenges that computational methods face when dealing with humanities problems (cf. Chapter 2) affect this work and highlight our solutions.

In order to cope with the limited **data set size**, we have employed linguistic analysis components to produce deep semantic discourse representations. This allows us to define different semantic similarity measures that can be used in combination in largely unsupervised alignment methods. Thus only a limited amount of tuning data is needed. Although the use of deep linguistic representations is not trivial either, it allows finding expressive structures without relying on a huge amount of redundancies in the data.

We have employed various domain adaptation techniques in order to cope with the special **text characteristics** of the descriptions of rituals. The supervised domain adaptation techniques make use of existing annotated corpora from other domains and require only small amounts of in-domain annotation, based on partially adapted existing annotation rules. The re-use of existing annotated corpora is a prerequisite for creating fine-grained linguistic representations, because large annotated data sets are needed as training material. We have not used statistical adaptation approaches for word sense disambiguation and coreference resolution and this is also due to data set limitations. Training statistical models for both tasks requires huge amount of training material, as supervised word sense disambiguation systems are usually trained per lemma (Navigli, 2009) and coreference resolution is a document/discourse-level phenomenon. In both cases, we have devised specific adaptation techniques that make use of domain-relevant data or domain phenomena.

We have evaluated the technical machinery, in particular the alignment algorithms, as far as possible. Given the small size of the gold standard, we performed an indirect **evaluation** with the clustering experiment. An interactive evaluation, in which researchers from the respective humanities areas are actually using produced analyses, would be an extrinsic evaluation, but this is hard to operationalize. The evaluation of the alignment algorithms against a manually constructed gold standard highlighted different strengths of Bayesian model merging and predicate alignment: The Bayesian model merging algorithm produces correct alignment links on pairs where the other algorithms fail to produce a single correct link. Predicate alignment achieves higher precision and recall scores on the other pair of documents.

In order to make the system outputs accessible and usable, we have developed **visualization** tools that show the output of the various components of the system. Each visual representation is linked with the underlying discourse representation and textual material. This allows the humanities researcher not only to find examples for publication, but also provides a means for verification against processing errors. We also described in Chapter 7 how a humanities researcher can use these tools to discover specific new areas of interest in an iterative process.

8.2 Contributions

The major contributions of this thesis fall in five areas:

Linguistic Processing and Discourse Representation We have described a modular processing architecture that produces fully integrated semantic representations of discourses. The discourse representations are based on automatic annotations from many linguistic levels, from part of speech to coreference chains. The representation scheme does not only contain the linguistic annotation layers in isolation. Instead, the annotation objects are linked to each other and can be exploited in conjunction and in their interaction. Technically, the XML data format we used is clearly defined and data files can be validated using XML schema.

Domain Adaptation The fact that most linguistic processing tools are developed for and trained on newspaper texts gives rise to the need for domain adaptation, because many texts used in the humanities are not newspaper texts. For each linguistic layer processed in our architecture, we have described techniques for the adaptation of the individual components to the ritual domain. Most of the techniques, however, can be employed similarly for the adaptation to other domains. The modularization of the processing architecture is a prerequisite for domain adaptation on individual linguistic levels. We focused on simple adaptation techniques that rely on retraining only, because they can easily be employed within DH projects. Also, we have shown that with small amounts of manually annotated data significant performance gains could be achieved with retraining.

Event Alignment Algorithms We have described three different algorithms for the alignment of events with different properties and have developed multi-factorial measures for semantic similarity of events. They exploit both semantic similarity of the event terms and the arguments and also consider relative distance as an important structural criterion. We applied the algorithms to folktales as well as descriptions of rituals in order to generate alignments.

For the evaluation of the alignment algorithms we established a gold standard of alignments in the ritual domain. Both Bayesian model merging and predicate alignment achieved a performance above the lemma alignment baseline, indicating the advantage of measuring similarity taking multiple factors into account. The Bayesian model merging has also produced a higher number of correct positive alignments over all story pairs, and can be considered most robust according to this evaluation. Predicate alignment achieves highest precision and recall scores on a single story pair.

The cluster evaluation, in which we compared clusterings induced by the alignment algorithms against a gold standard classification proved to be not fully reliable: A high number of imprecise alignment links can still produce correct clusters, without the individual links being correct indicators of structural similarities. This tendency can be seen from the strong performance of shallow similarity measures and different performance results for Needleman-Wunsch across the two experiments. Furthermore, the gold clustering provided by the ATU index might be more indicative of topical as opposed to structural similarity.

Identifying Event-level Similarities Based on the integrated discourse representations, we described a method to detect structural similarities of event sequences using alignment techniques. We align events across discourses by use of appropriate similarity functions for the alignment algorithms. We described how these alignments can be used to (i) quantify story similarity in general and (ii) detect specific similar elements in particular. We would like to point out that the alignment algorithms have been used on automatically pre-processed data that includes noise and processing errors. Nevertheless, we have shown how alignments can be visualized and used by humanities researchers. In addition, because the discourse representations contain arguments that have been connected by coreference chains, they offer many ways of analyzing narratives that go beyond events. Instead, the analysis can focus on characters and the events they participate in.

Visualization and Accessibility In digital humanities, the accessibility of results for researchers from the humanities is of utmost importance. Numeric evaluation scores, even if they are available, are difficult to interpret properly for researchers with a humanities background, because they lack the technical background. We have therefore shown how the results we have produced can be visualized and performed a showcase analysis on the descriptions of rituals.

8.3 Outlook and Future Work

As an outlook, we will discuss two areas that are, from our perspective, worth working on in the future. (i) Obviously, many components of the entire system can be improved.

We will discuss the most important ones and suggest some ideas. (ii) We have discussed ritual research and folkloristics as application scenarios, but it is our belief that the automatic detection of structural similarities can be of use in many more scenarios. We will describe some possibilities.

System improvements

Events A pressing issue is the notion of "event", or, from a technical standpoint, the input to the alignment algorithms. In the current setup, we use FrameNet frames as event representations and collect them as event sequences. Although many events are captured correctly by this approach, the input sequences also contain frames likes KINSHIP, which we clearly do not want there. They introduce noise into the alignment algorithms and have a bad influence on, e.g., the distance similarity measure. Long lists of family relatives appear sometimes in the descriptions of rituals. This causes the relative positions of the sequence elements around the list to be afar, although nothing really happened in terms of actions. A more restrictive preselection of events that are actually used in the event sequences could improve the alignment results. Relatively straightforward would be to use FrameNet frame inheritance and, e.g., allow only frames that inherit from the frame EVENT. In this regard, the special status of statives should also be taken into consideration.

Event similarity Another point that could be improved is the measuring of event similarity. Using different measures and averaging other them is relatively straightforward, but it is also very shallow because nuances that are captured by a single measure are removed due to the averaging. There are a number of cases in which a single measure would allow to make the correct choice, but it has not enough weight to overrule the other measures. Remember that the weights of the measures are fixed on a global level, i.e., the same weighting scheme is used for each pair. It would presumably improve the results if there was a way to decide the weight of the measures not globally, but per event pair or event pair type. If, for instance, both events are TEXT_CREATION-events, the similarity of the fillers of the frame elements becomes much more important than if one event is a PLACING and the other a CAUSE_FLUIDIC_MOTION. In order to do such a thing, we would require a classification of potential alignment links for which different weighting schemes could be used.

Characters In our current setup, characters come into play as arguments of events or entities created by the coreference resolution system. The similarity of characters across stories is measured only in terms of argument overlap. A more direct and explicit handling of characters, e.g., by inducing some sort of binding list as in Fay (2012), could be helpful to improve event alignments. However, one has to be careful to not being to restrictive, in particular if named entities are involved. Obviously, the similarity should not suffer, if Hansel and Gretel are named differently.

Connecting dense regions An obvious improvement to the random-walk algorithm to uncover densely connected regions across two aligned documents (Section 6.6) would be to base the algorithm on a weighted graph where the weights are given by the similarities of aligned events. This way, strongly connected regions would be receive a higher connectivity score if they link events that are similar. However, it is not directly clear what weight the sequential links in the graph should receive.

Application scenarios We have described ritual research and folkloristics as two application scenarios for this system: Both scholarly areas have an interest in event similarities across different narratives. However, we believe that there are many more application scenarios for identifying structural similarities.

Biographies Many areas in social sciences are interested in analyzing biographies (cf. Roberts, 2002). The *comparison* of biographies also has a long tradition, starting with Plutarch's "Parallel Lives", in which he made pairwise comparisons of Roman and Greek noblemen. If one would see a written biography as a story, a system like the one presented in this thesis would be able to uncover similarities in the story line, i.e., the lives of people. Although dates and locations are important in biographies, the similarities this system could detect go beyond that by taking the order and the relation to other persons into account (via coreference analyses).

Contemporary history Finding similarities across texts that describe temporal developments might be another interesting application scenario. Tanca (1993) described a number of cases in which international armed forces intervened in state-internal conflicts. Each intervention is usually preceded by discussions in the U.N. security council, possibly resolutions, etc. Finding similarities in the context of international interventions might help to identify key turning points in the process.

Summarization A system that identifies structural similarities in narrative texts, such as ours, might also be useful to improve a multi-document summarization (McKeown and Radev, 1995) application. In multi-document summarization, the task is to identify the key pieces of information from many documents about the same topic. Identifying similar events and participants across multiple documents allows such a system to detect important (and consistent) events and participants. Those can be chosen for generating a summary.

Appendix

1 Folktale: Bearskin

A soldier, having deserted his regiment in the thick of battle, took refuge in the woods. However, the foes of war were soon replaced by the enemies cold, thirst, and hunger. With nowhere to turn for help, he was about to surrender to the powers of despair, when without warning an awful spirit appeared before him. He offered the poor soldier great wealth, if he would but serve this uncanny master for seven years. Seeing no other escape from his misery, the soldier agreed.

The terms of the pact were quickly stated: For seven years the soldier was to wear only a bearskin robe, both day and night. He was to say no prayers. Neither comb nor shears were to touch his hair and beard. He was not to wash, nor cut his nails, nor blow his nose, nor even wipe his behind. In return, the spirit would provide him with tobacco, food, drink, and an endless supply of money.

The soldier, who by his very nature was not especially fond of either prayers or of cleanliness, entered into the agreement. He took lodgings in a village inn, and discovered soon enough that his great wealth was ample compensation for his strange looks and ill smell.

A nobleman frequented this inn. Impressed by Bearskin's lavish and generous expenditures, he presented him with a proposal. "I have three beautiful daughters," he said. "If the terms are right, you may choose any one of them for a bride."

Bearskin named a sum that was acceptable to the nobleman, and the two set forth to the palace to make the selection. The two older daughters made no attempt to hide their repugnance of the strange suitor, but the youngest unhesitatingly accepted her father's will. Bearskin formalized the betrothal by removing a ring from his own finger and twisting it into two pieces. One piece he gave to his future bride; the other he kept. Saying that soon he would return, he departed.

The seven years were nearly finished, so a short time later Bearskin did indeed come back for his bride. Now freshly bathed, neatly shorn, elegantly dressed, and riding in a luxurious carriage, he was a suitor worthy of a princess. Identifying himself with his half of the twisted ring, he claimed his bride.

Beside themselves with envy, and furious that they had squandered their rights to this handsome nobleman, one of the bride's older sisters hanged herself from a tree and the other one drowned herself in a well. Thus the devil gained two souls for the one that he had lost.

2 Proppian Event Functions

Symbol	Description
A	The villain causes harm or injury to a member of a family.
B	Misfortune or lack is made known; the hero is approached with a request or command; he is allowed to go or he is dispatched.
C	The seeker agrees to or decides upon counteraction.
↑	The hero leaves home.
D	The hero is tested, interrogated, attacked etc., which prepares the way for his receiving either a magical agent or helper.
E	The hero reacts to the actions of the future donor.
F	The hero acquires the use of a magical agent.
G	The hero is transferred, delivered or led to the whereabouts of an object of search.
H	The hero and the villain join in direct combat.
J	The hero is branded.
I	The villain is defeated.
K	The initial misfortune or lack is liquidated.
↓	The hero returns.
Pr	The hero is pursued.
Rs	Rescue of the hero from pursuit.
L	A false hero presents unfounded claims.
M	A difficult task is proposed to the hero.
N	The task is resolved.
Q	The hero is recognized.
Ex	The false hero or villain is exposed.
T	The hero is given a new appearance.
U	The villain is punished.
W	The hero is married and ascends the throne.

3 Description of a Cūḍākaraṇa Ritual

Salutation to Śrī Gaṇeśa.
Now the ritual of the first shaving of the head.
The yajamāna should sip three times water from the palm of the hand.
Place a plate with pūjā materials such as flowers etc. on the ground.
vākya starting with: "Today etc."
The Brahmin should perform the worship of the kalaśa with the siddhir astu…until yathāvāṇa.
Perform here the worship of the sixteen digits of the moon's disc on the bronze plate with salutations to Indra, Candra, Niśānātha, Śītāṃśu, Śaśalāñchana, Vidhu, Tārādhipati, Śaśin, Abja, Uḍupa, Ṛkṣa, Pūrṇimā and Dvijarāja.
Recitation of the imaṃ devā asupatnam.

Act here in the yathākarma.
The nāyaka should bring the boy holding his hand and make him sit on a svastika.
Fan the smoke of burnt rape and mustard seeds reciting the rakṣohanaṃ.
Wash ritually the body of the boy with water and rice reciting the adhy avoca.
Salutation.
Offer a lamp with a burning wick and the tejo 'si.
Sprinkle water from the arghyapātra reciting the devasya tvā.
Let the boy worship the sacred vase saying: "This seat is for all the filled sacred vases or the deities invoked in the vases".
Salutation.
Salutation with flowers.
Give a tikā to the yajamāna and/or boy with sandalwood paste and vermilion.
Salutation with flowers and a yajñopavīta.
Incense.
Light with a burning wick.
Now fragmant materials etc.
Worship of the lamp, the wooden measuring vessel and the key reciting the agnir mūrdhā divaḥ and the trātāram indram.
One should wave with lamp, wooden measuring vessel and key.
Offering of oil.
Wave a bamboo plate reciting the ausraghnam.
Offer oil on the head, hands and legs of the boy with the kāṇḍātkāṇḍāt.
The worshipper should comb the hair of the boy with a porcupine bristle and divide it into two parts reciting the dīrghāyutvāya.
Bind wood and leaves in the hair.
For it is said: "In the east above the forehead, a piece of the bar.
In the south above the right ear, a piece of the dubasi, on the left i.e. north, above the left ear, a piece of the valasi, in the west also above the right ear, a piece of the bastard teak or flame of the palasi."
The following is the oṣadhe trāyasva for binding the wood and leaves into the hair.
After this draw a svastika on the hands of the maternal uncle and worship the hands.
Give dakṣiṇā to the priest or gods.
Hand over a golden needle, a silver needle, a golden razor and a silver razor to the maternal uncle.
The father should pour hot and/or cold water reciting the uṣṇena vāya.
By this mikhiścāpa.
The father should pour water in the east of the hair, then should the maternal uncle shave the hair at the given auspicious moment reciting the ya bhūriścarā divaṃ.
The same in the south reciting the oṣadhe trāyasva svadhite mainam himsīḥ.
The same in the north reciting the śivo nāmāmsi.
The same in the west reciting the ya bhūriścarā divaṃ.
Imagine that the whole head is shaved reciting the yatkṣureṇa māskāyu mukhaniṣī.
Recite the mūrdhānaṃ divo aratiṃ.
Pierce the ears: on the right side with a golden needle, on the left side with a silver

needle reciting the bhadraṃ karṇebhiḥ śṛṇuyāma.
Give sandalwood paste etc. and svagã.
Shower pieces of fruits etc. from the measuring vessel on the head of the boy with the yāḥ phalini.
Make this three times.
Show and offer the lamp to the boy with the tejo 'si.
Everybody should throw popped rice on the head of the boy while the priest recites the mano jūtir.
Worship the hands of the barber.
Give the golden and silver razor with dakṣiṇā to the barber.
Give him also a small plate.
After finishing this much, the nāyaḥ should take away the boy holding his hand.
Place him on the seat decorated with a svastika.
Shave the head.
The nini should collect the shaved hair.
Throw sweet meat on the plate for the barber.
Let the boy be besmeared and bath with mustard oil cake.
Let the boy undress.
After finishing this, the nāyaḥ should bring the boy holding his hand.
Let the boy again sit on the seat decorated with a svastika.
Fan the smoke of burnt rape and mustard seeds reciting the rakṣohanaṃ.
Clean the eyes with uncooked rice and water and place the rice in the woven bamboo basket reciting the adhy avocad.
Show and offer the lamp to the boy with the tejo 'si.
Sprinkle water from the arghyapātra reciting the devasya tvā.
Let the boy worship the sacred vase saying: "This seat is for all the filled sacred vases".
Salutation.
Salutation with flowers.
Also give a tikā of sandalwood paste and vermilion to the boy.
Give him the yajñopavīta.
Burn incense.
Wave light with a burning wick.
Now fragrant materials etc.
Worship the lamp, the wooden measuring vessel and the key reciting the agnir mūrdhā and the trātāram indram.
Wave the lamp, the wooden measuring vessel and the iron key over the head of the boy reciting the ausraghnam.
Draw on the head of the boy a svastika with sandalwood paste.
Apply this sandalwood paste on the whole head reciting the gandhadvārāṃ.
Place some white sesame on the head of the boy.
Bind the kumaḥkaḥ around on the head with the rakṣohanaṃ.
Bind a silk thread around the head with the pavitre 'stho.
Stick a porcupine bristle, stick a traditional comb, stick a piece of kuśa in the hair again with the pavitre 'stho.

For kuśa the brahmanaspate.
Apply black soot on the eyes of the boy reciting yuñjanti bradhnam.
Bind a phani on the tuft reciting tava vāyav.
Wave the thāybhū on which is a candramaṇḍala is drawn.
Give a svagã.
Offer rice to the gods.
Paste a tikā of sandalwood paste on the forehead of the child.

4 Mathematical Notation Overview

Representations	$\mathcal{S}$	Set of sequences
	$\mathcal{A}$	Set of possible alignments
	$\mathcal{E}$	Set of events
	$\mathcal{M}$	Set of hidden Markov models
	$S \in \mathcal{S}, S \subseteq \mathcal{E}, S = \langle s_0, s_1, \ldots s_n \rangle$	An ordered set of events, a sequence
	$T \in \mathcal{S}, T \subseteq \mathcal{E}, T = \langle t_0, t_1, \ldots t_m \rangle$	
	$A_{S,T} \in \mathcal{A}$	Set of alignment links over S and T
	$\lvert S \rvert = \lvert \langle s_0, s_1, \ldots, s_n \rangle \rvert = n$	Number of sequence elements, length of a sequence
	$\langle a_i \rangle_0^k = \langle a_0, a_1, \ldots, a_k \rangle$	Short notation for sequences
	$s \in S, t \in T$	Single sequence elements, single events
	$s_1 \rightarrow s_2$	Sequence element s_2 directly follows element s_1
	$a \in A_{S,T}, a \subseteq (S \cup T)$	A single alignment link, set of aligned sequence elements
Functions	$\mathrm{g} : \mathbb{N} \rightarrow \mathbb{R}$	Gap cost function (Needleman-Wunsch, p. 65)
	$\mathrm{geo} : \mathcal{M} \rightarrow [0,1]$	Geometric function (Bayesian model merging, p. 69)
	$\mathrm{plaus} : \mathcal{M} \rightarrow \{0,1\}$	Plausibility function (Bayesian model merging, p. 69)
	$\mathrm{sim} : \mathcal{E} \times \mathcal{E} \rightarrow [0,1]$	Similarity of individual events (p. 73)
	$\mathrm{sim} : A_{S,T} \rightarrow [0,1]$	Similarity within an alignment link (p. 83)
	$\mathrm{sim}_{\mathrm{doc}} : S \times T \times A_{S,T} \rightarrow [0,1]$	Document/sequence similarity (p. 83)
	$\mathrm{rwalk} : G \times V \times k \rightarrow V^k$	Random walk, returns a sequence of k events (p. 88)
	$\mathrm{c} : G \times V \times k \rightarrow [0,k]$	Connectivity score based on one random walk; absolute frequency of crossing the sequences (p. 88)

5 Discourse Representation File Format

5.1 XML Schema

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema elementFormDefault="unqualified" attributeFormDefault="unqualified"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:xs="http://www.w3.org/2001/XMLSchema-instance">
  <xsd:element name="root">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="document" type="documentType"></xsd:element>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
  <xsd:complexType name="documentType">
    <xsd:sequence>
      <xsd:element name="originaltext" type="xsd:string" maxOccurs="1"
          minOccurs="1">
      </xsd:element>
      <xsd:element name="sentences" maxOccurs="1" minOccurs="1">
        <xsd:complexType>
          <xsd:sequence>
            <xsd:element name="sentence" type="sentenceType"
                maxOccurs="unbounded" minOccurs="0">
            </xsd:element>
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
      <xsd:element name="coreference" maxOccurs="1" minOccurs="1">
        <xsd:complexType>
          <xsd:sequence>
            <xsd:element name="entity" type="entityType"
                maxOccurs="unbounded" minOccurs="0">
            </xsd:element>
            <xsd:element name="singletons" maxOccurs="1"
                minOccurs="0">
              <xsd:complexType>
                <xsd:sequence>
                  <xsd:element name="mention" type="mentionType"
                      maxOccurs="unbounded" minOccurs="0" />
                </xsd:sequence>
              </xsd:complexType>
            </xsd:element>
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
      <xsd:element name="frames" maxOccurs="1" minOccurs="1">
        <xsd:complexType>
          <xsd:sequence>
            <xsd:element name="frame" type="frameType"
                maxOccurs="unbounded" minOccurs="0" />
            <xsd:element name="order" type="orderType" maxOccurs="1"
                minOccurs="1" />
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
      <xsd:element name="chunks" maxOccurs="1" minOccurs="1">
        <xsd:complexType>
          <xsd:sequence>
            <xsd:element name="chunk" type="chunkType"
                maxOccurs="unbounded" minOccurs="0" />
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
      <xsd:element name="sections" maxOccurs="1" minOccurs="1">
        <xsd:complexType>
          <xsd:sequence>
            <xsd:element name="section" type="sectionType"
                maxOccurs="unbounded" minOccurs="0" />
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
      <xsd:element name="senses" maxOccurs="1" minOccurs="1">
        <xsd:complexType>
          <xsd:sequence>
            <xsd:element name="sense" type="senseType"
                maxOccurs="unbounded" minOccurs="0" />
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
      <xsd:element name="mantras" maxOccurs="1" minOccurs="0">
        <xsd:complexType>
          <xsd:sequence>
            <xsd:element name="mantra" type="mantraType" maxOccurs="unbounded"
                minOccurs="0" />
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
    </xsd:sequence>
    <xsd:attribute ref="id" />
  </xsd:complexType>
  <xsd:complexType name="sentenceType">
    <xsd:sequence>
      <xsd:element name="token" type="tokenType" maxOccurs="unbounded"
          minOccurs="1" />
    </xsd:sequence>
    <xsd:attribute ref="id" />
  </xsd:complexType>
  <xsd:complexType name="tokenType">
    <xsd:sequence>
      <xsd:element name="frame" type="frameRefType"
          maxOccurs="unbounded" minOccurs="0" />
    </xsd:sequence>
    <xsd:attribute name="word" type="xsd:string" />
    <xsd:attribute name="lemma" type="xsd:string" use="required" />
    <xsd:attribute name="sense" type="xsd:IDREF" />
    <xsd:attribute name="characterOffsetBegin" type="xsd:int" />
    <xsd:attribute name="characterOffsetEnd" type="xsd:int"
        use="required" />
    <xsd:attribute name="governor" type="xsd:IDREF" />
    <xsd:attribute name="deprel" type="xsd:string" />
    <xsd:attribute ref="id" />
    <xsd:attribute name="pos" type="xsd:string" use="required" />
    <xsd:attribute ref="OldId" />
  </xsd:complexType>
  <xsd:complexType name="frameType">
    <xsd:sequence>
```

```
    <xsd:element name="token" type="tokenRefType" maxOccurs="1"
        minOccurs="1" />
    <xsd:element name="frame_element" maxOccurs="unbounded"
        minOccurs="0" />
      <xsd:complexType>
        <xsd:sequence maxOccurs="unbounded" minOccurs="1">
          <xsd:choice maxOccurs="unbounded" minOccurs="1">
            <xsd:element name="mention" type="mentionRefType"
                maxOccurs="unbounded" minOccurs="0" />
            <xsd:element name="token" type="tokenRefType"
                maxOccurs="unbounded" minOccurs="0" />
            <xsd:element name="head" type="tokenRefType"
                maxOccurs="1" minOccurs="0" />
          </xsd:choice>
        </xsd:sequence>
        <xsd:attribute name="name" type="xsd:string" />
        <xsd:attribute ref="id" />
      </xsd:complexType>
    </xsd:element>
  </xsd:sequence>
  <xsd:attribute name="name" type="xsd:string" />
  <xsd:attribute ref="id" />
  <xsd:attribute ref="OldId" />
</xsd:complexType>
<xsd:complexType name="tokenRefType">
  <xsd:attribute name="idref" type="xsd:IDREF" />
</xsd:complexType>
<xsd:complexType name="chunkType">
  <xsd:sequence>
    <xsd:element name="token" type="tokenRefType"
        maxOccurs="unbounded" minOccurs="1" />
  </xsd:sequence>
  <xsd:attribute name="category" type="xsd:string" />
  <xsd:attribute ref="id" />
  <xsd:attribute name="sentence" type="xsd:IDREF" />
</xsd:complexType>
<xsd:complexType name="senseType">
  <xsd:attribute name="wordnet" type="xsd:string" use="required" />
  <xsd:attribute ref="id" />
</xsd:complexType>
<xsd:complexType name="sectionType">
  <xsd:sequence>
    <xsd:element name="sentence" type="sentenceRefType"
        maxOccurs="unbounded" minOccurs="0" />
  </xsd:sequence>
  <xsd:attribute ref="id" />
</xsd:complexType>
<xsd:attribute name="id" type="xsd:ID" />
<xsd:complexType name="entityType">
  <xsd:sequence>
    <xsd:element name="sense" type="senseRefType"
        maxOccurs="1" minOccurs="0" />
    <xsd:element name="mention" type="mentionType"
        maxOccurs="unbounded" minOccurs="1" />
    </xsd:sequence>
    <xsd:attribute ref="id" />
  </xsd:complexType>
  <xsd:complexType name="senseRefType">
    <xsd:attribute name="idref" type="xsd:IDREF" />
  </xsd:complexType>
  <xsd:complexType name="mentionType">
    <xsd:sequence>
      <xsd:element name="token" type="tokenRefType"
          maxOccurs="unbounded" minOccurs="1" />
      <xsd:element name="fe" type="frameElementRefType"
          maxOccurs="unbounded" minOccurs="0" />
    </xsd:sequence>
    <xsd:attribute ref="id" />
  </xsd:complexType>
  <xsd:complexType name="frameElementRefType">
    <xsd:attribute name="idref" type="xsd:IDREF" />
  </xsd:complexType>
  <xsd:complexType name="mentionRefType">
    <xsd:attribute name="idref" type="xsd:IDREF" />
  </xsd:complexType>
  <xsd:complexType name="frameRefType">
    <xsd:attribute name="idref" type="xsd:IDREF" />
  </xsd:complexType>
  <xsd:complexType name="orderType">
    <xsd:sequence>
      <xsd:element name="frame" type="frameRefType"
          maxOccurs="unbounded" minOccurs="0" />
    </xsd:sequence>
    <xsd:attribute name="type">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="temporal" />
          <xsd:enumeration value="textual" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:attribute>
  </xsd:complexType>
  <xsd:complexType name="sentenceRefType">
    <xsd:attribute name="idref" type="xsd:IDREF"></xsd:attribute>
  </xsd:complexType>
  <xsd:complexType name="mantraType">
    <xsd:sequence>
      <xsd:element name="token" type="tokenRefType"
          maxOccurs="unbounded" minOccurs="1" />
    </xsd:sequence>
    <xsd:attribute name="id" type="xsd:ID"/>
  </xsd:complexType>
  <xsd:attribute name="OldId" type="xsd:string" />
</xsd:schema>
```

5.2 XML Example

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<root>
  <document id="r0009">
    <originaltext><![CDATA[Salutation to Śrī Gaṇeśa.
Now the ritual of the first shaving of the head.
[...]
  ]]></originaltext>
    <sentences>
      <sentence id="s0">
        <token id="t722" word="Salutation" lemma="salutation"
          characterOffsetBegin="0" characterOffsetEnd="10" pos="NN" OldId="r0009_0_t_0">
          <frame idref="f0"/>
        </token>
        <token id="t677" word="to" lemma="to" deprel="PREP"
          characterOffsetBegin="11" characterOffsetEnd="13"
          governor="t722" pos="TO" OldId="r0009_0_t_1"/>
        <token id="t666" word="Śrī" lemma="Śrī" deprel="NN"
          characterOffsetBegin="14" characterOffsetEnd="17"
          governor="t688" pos="NNP" OldId="r0009_0_t_2"/>
        <token id="t688" word="Gaṇeśa" lemma="Gaṇeśa" deprel="POBJ"
          characterOffsetBegin="18" characterOffsetEnd="24"
          governor="t677" pos="NNP" OldId="r0009_0_t_3"/>
        <token id="t780" word="." lemma="." deprel="PUNCT"
          characterOffsetBegin="24" characterOffsetEnd="25"
          governor="t722" pos="." OldId="r0009_0_t_4"/>
      </sentence>
      [...]
    </sentences>
    <coreference>
      <entity id="e3">
        <mention id="m12">
          <token idref="t829"/>
          <token idref="t840"/>
          <token idref="t865"/>
        </mention>
        <mention id="m14">
          <token idref="t669"/>
          <token idref="t716"/>
          <token idref="t730"/>
          <fe idref="fe98"/>
          <fe idref="fe99"/>
        </mention>
        <mention id="m17">
          <token idref="t235"/>
          <token idref="t246"/>
          <token idref="t263"/>
          <fe idref="fe113"/>
          <fe idref="fe115"/>
          <fe idref="fe111"/>
        </mention>
        [...]
      </entity>
    </coreference>
    <frames>
      [...]
      <frame id="f3" OldId="r00092_f0" name="Ingestion">
        <token idref="t895"/>
        <frame_element id="fe6" name="Source">
          <head idref="t986"/>
          <token idref="t962"/>
          <token idref="t976"/>
          <token idref="t986"/>
          <token idref="t1054"/>
          <token idref="t940"/>
          <token idref="t955"/>
        </frame_element>
        <frame_element id="fe4" name="Ingestor">
          <head idref="t6"/>
          <token idref="t6"/>
          <token idref="t928"/>
          <mention idref="m0"/>
        </frame_element>
        <frame_element id="fe5" name="Ingestibles">
          <head idref="t997"/>
          <token idref="t997"/>
        </frame_element>
      </frame>
      [...]
    </frames>
    <chunks>
      <chunk id="c2" category="NP" sentence="s0">
        <token idref="t688"/>
        <token idref="t666"/>
      </chunk>
      [...]
    </chunks>
    <senses>
      <sense id="sen149" wordnet="601611-v"/>
      <sense id="sen147" wordnet="14373933-n"/>
      [...]
    </senses>
    <mantras>
      <mantra id="mantra0">
        <token idref="t23"/>
      </mantra>
      [...]
    </mantras>
  </document>
```

Bibliography

Aarne, Antti and Stith Thompson. *The Types of the Folktale*. 2nd. Vol. 75. FF Communications 184. Helsinki, Finland: Suomalainen Tiedeakatemia, May 1961.

Agirre, Eneko and Aitor Soroa. Personalizing PageRank for Word Sense Disambiguation. In: *Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009)*. Ed. by Alex Lascarides, Claire Gardent, and Joakim Nivre. Athens, Greece: Association for Computational Linguistics, Mar. 2009, pp. 33–41.

Andel, Kevin M. van. Formalizing TV Crime Series: Application and Evaluation of the Doxastic Preference Framework. Bachelor's thesis. University of Amsterdam, 2010.

Apache Software Foundation. *UIMA*. URL: `http://uima.apache.org` (visited on 02/10/2014).

Arslan, Hamdiye. Temporale Annotation narrativer Texte: Vergleich zwischen Fabeltexten und Ritualtexten. Bachelor's thesis. Heidelberg University, 2013.

Ashliman, D. L. *A Guide to Folktales in the English Language: Based on the Aarne-Thompson Classification System*. Vol. 11. Bibliographies and Indexes in World Literature. Westport, New York and London: Greenwood Press, 1987.

– *Folktexts: A library of folktales, folklore, fairy tales, and mythology*. University of Pittsburgh. 1996. URL: `http://www.pitt.edu/~dash/folktexts.html` (visited on 02/10/2014).

Bagga, Amit and Breck Baldwin. Algorithms for Scoring Coreference Chains. In: *Proceedings of the Workshop on Linguistic Coreference held at the First International Conference on Language Resources and Evaluation;* Granada, Spain, May 1998.

Banerjee, Satanjeev and Ted Pedersen. An Adapted Lesk Algorithm for Word Sense Disambiguation Using WordNet. In: *Computational Linguistics and Intelligent Text Processing*. Ed. by Alexander Gelbukh. Vol. 2276. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2002, pp. 136–145.

Beißwenger, Michael and Angelika Storrer. Corpora of Computer-Mediated Communication. In: *Corpus Linguistics. An International Handbook.* Ed. by Anke Lüdeling and Merja Kytö. Vol. 2. Handbooks of Linguistics and Communication Science. Berlin: Mouton De Gruyter, 2009.

Blitzer, John, Mark Dredze, and Fernando Pereira. Biographies, Bollywood, Boomboxes and Blenders: Domain Adaptation for Sentiment Classification. In: *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*. Ed. by Annie Zaenen and Antal van den Bosch. Prague, Czech Republic: Association for Computational Linguistics, June 2007, pp. 440–447.

Blitzer, John, Ryan McDonald, and Fernando Pereira. Domain Adaptation with Structural Correspondence Learning. In: *Proceedings of the 2006 Conference on Empirical*

Methods in Natural Language Processing. Ed. by Dan Jurafsky and Eric Gaussier. Sydney, Australia: Association for Computational Linguistics, July 2006, pp. 120–128.

Bod, Rens, Bernhard Fisseni, Aadil Kurji, and Benedikt Löwe. Objectivity and Reproducibility of Proppian Narrative Annotations. In: *Proceedings of the Third Workshop on Computational Models of Narrative*. Ed. by Mark Alan Finlayson. May 2012, pp. 17–21.

Bohnet, Bernd. Top Accuracy and Fast Dependency Parsing is not a Contradiction. In: *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*. Ed. by Chu-Ren Huang and Dan Jurafsky. Beijing, China: Coling 2010 Organizing Committee, Aug. 2010, pp. 89–97.

Bollobás, Béla. *Modern Graph Theory*. Vol. 184. Graduate Texts in Mathematics. Springer Berlin / Heidelberg, 1998.

Boyd-Graber, Jordan, Christiane Fellbaum, Daniel Osherson, and Robert Schapire. Adding Dense, Weighted Connections to WordNet. In: *Proceedings of the Third International WordNet Conference*. Ed. by Petr Sojka, Key-Sun Choi, Christine Fellbaum, and Piek Vossen. Jeju Island, Korea, Jan. 2006, pp. 29–35.

Brooke, Julian, Graeme Hirst, and Adam Hammond. Clustering Voices in The Waste Land. In: *Proceedings of the Workshop on Computational Linguistics for Literature*. Ed. by David Elson, Anna Kazantseva, and Stan Szpakowicz. Atlanta, Georgia: Association for Computational Linguistics, June 2013, pp. 41–46.

Brosius, Christiane, Axel Michaels, and Paula Schrode, eds. *Ritual und Ritualdynamik*. Göttingen, Germany: Vandenhoeck & Ruprecht, 2013.

Buchholz, Sabine. *chunklink*. 2000. URL: `http://www.cnts.ua.ac.be/conll2000/chunking/` (visited on 02/10/2014).

Burchardt, Aljoscha, Marco Pennacchiotti, Stefan Thater, and Manfred Pinkal. Assessing the impact of frame semantics on textual entailment. In: *Natural Language Engineering* 15, Special Issue 4 Sept. 2009, pp. 527–550.

Busa, Roberto. The Annals of Humanities Computing: The Index Thomisticus. In: *Computers and the Humanities* 14, 1980, pp. 83–90.

Byrnes, Robert. A statistical analysis of the "Eumaeus" Phrasemes in James Joyce's Ulysses. In: *Proceedings of the 10th International Conference on Statistical Analysis of Textual Data*. Ed. by Bolasco Sergio, Chiari Isabella, and Giuliano Luca. Rome, Italy: LED Edizioni Universitarie, June 2010.

Cai, Jie and Michael Strube. Evaluation Metrics For End-to-End Coreference Resolution Systems. In: *Proceedings of the SIGDIAL 2010 Conference*. Ed. by Raquel Fernández, Yasuhiro Katagiri, Kazunori Komatani, Oliver Lemon, and Mikio Nakano. Tokyo, Japan: Association for Computational Linguistics, Sept. 2010, pp. 28–36.

Caliński, Tadeusz and Joachim Harabasz. A dendrite method for cluster analysis. In: *Communications in Statistics - Theory and Methods* 3 (1), 1974, pp. 1–27.

Camp, Matje van de and Antal van den Bosch. The socialist network. In: *Decision Support Systems* 53 (4), 2012, pp. 761–769.

Cer, Daniel, Marie-Catherine de Marneffe, Daniel Jurafsky, and Christopher D. Manning. Parsing to Stanford Dependencies: Trade-offs between speed and accuracy. In: *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*. Ed. by Nicoletta Calzolari, Khalid Choukri, Bente Maegaard, Joseph Mariani, Jan Odijk, Stelios Piperidis, Mike Rosner, and Daniel Tapias. Valletta, Malta: European Language Resources Association (ELRA), May 2010.

Chambers, Nathanael William. Inducing Event Schemas and their Participants from Unlabeled Texts. PhD thesis. Stanford University, May 2011.

Chan, Yee Seng and Hwee Tou Ng. Domain Adaptation with Active Learning for Word Sense Disambiguation. In: *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*. Ed. by Annie Zaenen and Antal van den Bosch. Prague, Czech Republic: Association for Computational Linguistics, June 2007, pp. 49–56.

Chatman, Seymour Benjamin. *Story and discourse: Narrative Structure in Fiction and Film*. Cornell University Press, 1980.

Clement, Tanya E. 'A thing not beginning and not ending': using digital tools to distant-read Gertrude Stein's *The Making of Americans*. In: *Literary and Linguistic Computing* 23 (3), 2008, pp. 361–381.

Cohen, Jacob. A Coefficient of Agreement for Nominal Scales. In: *Educational and Psychological Measurement* 20 (1), 1960, pp. 37–46.

Cook, Paul and Graeme Hirst. Automatically Assessing Whether a Text Is Clichéd, with Applications to Literary Analysis. In: *Proceedings of the 9th Workshop on Multiword Expressions*. Ed. by Valia Kordoni, Carlos Ramisch, and Aline Villavicencio. Atlanta, Georgia, USA: Association for Computational Linguistics, June 2013, pp. 52–57.

Correira, Alfred. Computing Story Trees. In: *American Journal of Computational Linguistics* 6 (3-4), July 1980, pp. 135–149.

Cybulska, Agata Katarzyna and Piek Vossen. Historical Event Extraction from Text. In: *Proceedings of the 5th ACL-HLT Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities*. Ed. by Kalliopi Zervanou and Piroska Lendvai. Portland, OR, USA: Association for Computational Linguistics, June 2011, pp. 39–43.

Das, Dipanjan, Nathan Schneider, Desai Chen, and Noah A. Smith. Probabilistic Frame-Semantic Parsing. In: *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. Ed. by Ron Kaplan, Jill Burstein, Mary Harper, and Gerald Penn. Los Angeles, California: Association for Computational Linguistics, June 2010, pp. 948–956.

Daumé III, Hal. Frustratingly Easy Domain Adaptation. In: *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*. Ed. by Annie Zaenen and Antal van den Bosch. Prague, Czech Republic: Association for Computational Linguistics, June 2007, pp. 256–263.

Dijkstra, Edsger Wybe. A note on two problems in connexion with graphs. English. In: *Numerische Mathematik* 1 (1), 1959, pp. 269–271.

Dipper, Stefanie. Morphological and Part-of-Speech Tagging of Historical Language Data: A Comparison. In: *Journal for Language Technology and Computational Linguistics* 26 (2), 2011, pp. 25–37.

Elson, David K. DramaBank: Annotating Agency in Narrative Discourse. In: *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*. Ed. by Nicoletta Calzolari, Khalid Choukri, Thierry Declerck, Mehmet Uğur Doğan, Bente Maegaard, Joseph Mariani, Jan Odijk, and Stelios Piperidis. Istanbul, Turkey: European Language Resources Association (ELRA), May 2012.

Elson, David K. Modeling Narrative Discourse. PhD thesis. Columbia University, New York City, 2012.

Elson, David K., Nicholas Dames, and Kathleen McKeown. Extracting Social Networks from Literary Fiction. In: *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*. Ed. by Jan Hajič, Sandra Carberry, Stephen Clark, and Joakim Nivre. Uppsala, Sweden: Association for Computational Linguistics, July 2010, pp. 138–147.

Fay, Matthew P. Story Comparison via Simultaneous Matching and Alignment. In: *Proceedings of the Third Workshop on Computational Models of Narrative*. Ed. by Mark Alan Finlayson. May 2012.

Fillmore, Charles J., Christopher R. Johnson, and Miriam R.L. Petruck. Background to FrameNet. In: *International Journal of Lexicography* 16 (3), 2003, pp. 235–250.

Finlayson, Mark Alan. Learning Narrative Structure from Annotated Folktales. PhD thesis. Massachusetts Institute of Technology, Feb. 2012.

Forster, Edward Morgan. *Aspects of the Novel*. London: Edward Arnold, 1927.

Frank, Anette, Thomas Bögel, Oliver Hellwig, and Nils Reiter. Semantic Annotation for the Digital Humanities. In: *Linguistic Issues in Language Technology* 7 (1), Jan. 2012.

Fraser, Alexander and Daniel Marcu. Measuring Word Alignment Quality for Statistical Machine Translation. In: *Computational Linguistics* 33 (3), 2007, pp. 293–303.

Goldberg, Andrew V. and Robert E. Tarjan. A new approach to the maximum-flow problem. In: *Journal of the ACM* 35 (4), Oct. 1988, pp. 921–940.

Goyal, Amit, Ellen Riloff, and Hal Daumé III. Automatically Producing Plot Unit Representations for Narrative Text. In: *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*. Ed. by Hang Li and Lluís Màrquez. Cambridge, MA: Association for Computational Linguistics, Oct. 2010, pp. 77–86.

Gutschow, Niels and Axel Michaels. *Handling Death. The Dynamics of Death and Ancestor Rituals Among the Newars of Bhaktapur*. Vol. 3. Ethno-Indology. Heidelberg Studies in South Asian Rituals. Wiesbaden: Harrassowitz Verlag, 2005.

– *Growing Up. Hindu and Buddhist Initiation Rituals among Newar Children in Bhaktapur*. Vol. 6. Ethno-Indology. Heidelberg Studies in South Asian Rituals. Wiesbaden: Harrassowitz Verlag, 2008.

Hellwig, Oliver. *DCS - The Digital Corpus of Sanskrit*. Heidelberg University. 2010. URL: `http://kjc-fs-cluster.kjc.uni-heidelberg.de/dcs/` (visited on 02/10/2014).

Hinrichs, Erhard W., Marie Hinrichs, and Thomas Zastrow. WebLicht: Web-Based LRT Services for German. In: *Proceedings of the ACL 2010 System Demonstrations*. Ed. by Sandra Kübler. Uppsala, Sweden: Association for Computational Linguistics, July 2010, pp. 25–29.

Holyoak, Keith J. and Paul Thagard. Analogical mapping by constraint satisfaction. In: *Cognitive Science* 13 (3), 1989, pp. 295–355.

Hovy, Eduard, Mitchell Marcus, Martha Palmer, Lance Ramshaw, and Ralph Weischedel. OntoNotes: The 90% Solution. In: *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers*. Ed. by Robert C. Moore, Jeff

Bilmes, Jennifer Chu-Carroll, and Mark Sanderson. New York City, USA: Association for Computational Linguistics, June 2006, pp. 57–60.

Hubert, Lawrence and Phipps Arabie. Comparing partitions. In: *Journal of Classification* 2 (1), 1985, pp. 193–218.

Inaki, Akiko and Tomoko Okita. A Small-Corpus-Based Approach to Alice's Roles. In: *Literary and Linguistic Computing* 21 (3), 2006, pp. 283–294.

Jiang, Jay J. and David W. Conrath. Semantic Similarity Based on Corpus Statistics and Lexical Taxonomy. In: *Proceedings of the 10th Research on Computational Linguistics International Conference*. 1997.

Jiang, Jing and ChengXiang Zhai. Instance Weighting for Domain Adaptation in NLP. In: *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*. Ed. by Annie Zaenen and Antal van den Bosch. Prague, Czech Republic: Association for Computational Linguistics, June 2007, pp. 264–271.

Jockers, Matthew L., Daniela M. Witten, and Craig S. Criddle. Reassessing authorship of the *Book of Mormon* using delta and nearest shrunken centroid classification. In: *Literary and Linguistic Computing* 23 (4), 2008, pp. 465–491.

Kao, Justine and Dan Jurafsky. A Computational Analysis of Style, Affect, and Imagery in Contemporary Poetry. In: *Proceedings of the NAACL-HLT 2012 Workshop on Computational Linguistics for Literature*. Ed. by David K. Elson, Anna Kazantseva, Rada Mihalcea, and Stan Szpakowicz. Montréal, Canada: Association for Computational Linguistics, June 2012, pp. 8–17.

Kawahara, Daisuke and Kiyotaka Uchimoto. Learning Reliability of Parses for Domain Adaptation of Dependency Parsing. In: *Proceedings of the Third International Joint Conference on Natural Language Processing*. Vol. 2. Asian Federation of Natural Language Processing. Jan. 2008.

Krifka, Manfred, Francis Jeffry Pelletier, Gregory N. Carlson, Alice ter Meulen, Gennaro Chierchia, and Godehard Link. Genericity: An Introduction. In: *The Generic Book*. Ed. by Gregory Norman Carlson and Francis Jeffry Pelletier. Chicago: University of Chicago Press, 1995. Chap. 1, pp. 1–124.

Kumar, Abhishek, Avishek Saha, and Hal Daume. Co-regularization Based Semi-supervised Domain Adaptation. In: *Advances in Neural Information Processing Systems 23*. Ed. by J.D. Lafferty, C.K.I. Williams, J. Shawe-Taylor, R.S. Zemel, and A. Culotta. 2010, pp. 478–486.

Lang, Andrew, ed. *The Blue Fairy Book*. Vol. 1. Fairy Books. Flying Chipmunk Publishing, 1889.

Lawson, E. Thomas and Robert McCauley. The cognitive representation of religious ritual form: A theory of participants' competence with religious ritual systems. In: *Current Approaches in the Cognitive Science of Religion*. Ed. by Ilkka Pyysiainen and Veikko Anttonen. New York: Continuum, 2002. Chap. 8, pp. 153–176.

Lehnert, Wendy G. Plot Units and Narrative Summarization. In: *Cognitive Science* 5 (4), 1981, pp. 293–331.

Lenat, Douglas B. CYC: a large-scale investment in knowledge infrastructure. In: *Communications of the ACM* 38 (11), 1995, pp. 33–38.

Liang, Percy, Ben Taskar, and Dan Klein. Alignment by Agreement. In: *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers*. Ed. by Robert C. Moore, Jeff Bilmes, Jennifer Chu-Carroll, and Mark Sanderson. New York City, USA: Association for Computational Linguistics, June 2006, pp. 104–111.

Lin, Dekang. Automatic Retrieval and Clustering of Similar Words. In: *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics*. Montreal, Quebec, Canada: Association for Computational Linguistics, Aug. 1998, pp. 768–774.

Löwe, Benedikt and Eric Pacuit. An abstract approach to reasoning about games with mistaken and changing beliefs. In: *Australasian Journal of Logic* 6, 2008, pp. 162–181.

Mani, Inderjeet. *Computational Modeling of Narrative*. Ed. by Graeme Hirst. Vol. 5. Synthesis Lectures on Human Language Technologies 3. Morgan & Claypool Publishers, Dec. 2012, pp. 1–142.

Manning, Christopher D. and Hinrich Schütze. *Foundations of Statistical Natural Language Processing*. Cambridge, Massachusetts and London, England: MIT Press, 1999.

Marcus, Mitchell P., Beatrice Santorini, Mary Ann Marcinkiewicz, and Ann Taylor. *Treebank-3*. Linguistic Data Consortium, Philadelphia. 1999. URL: `http://catalog.ldc.upenn.edu/LDC99T42` (visited on 01/14/2014).

Margolis, Anna, Karen Livescu, and Mari Ostendorf. Domain Adaptation with Unlabeled Data for Dialog Act Tagging. In: *Proceedings of the 2010 Workshop on Domain Adaptation for Natural Language Processing*. Ed. by Hal Daumé III, Tejaswini Deoskar, David McClosky, Barbara Plank, and Jörg Tiedemann. Uppsala, Sweden: Association for Computational Linguistics, July 2010, pp. 45–52.

Marneffe, Marie-Catherine de and Christopher D. Manning. The Stanford typed dependencies representation. In: *Coling 2008: Proceedings of the workshop on Cross-Framework and Cross-Domain Parser Evaluation*. Ed. by Johan Bos, Edward Briscoe, Aoife Cahill, John Carroll, Stephen Clark, Ann Copestake, Dan Flickinger, Josef van Genabith, Julia Hockenmaier, Aravind Joshi, Ronald Kaplan, Tracy Holloway King, Sandra Kuebler, Dekang Lin, Jan Tore Loenning, Christopher Manning, Yusuke Miyao, Joakim Nivre, Stephan Oepen, Kenji Sagae, Nianwen Xue, and Yi Zhang. Manchester, UK: Coling 2008 Organizing Committee, Aug. 2008.

McCarthy, Diana, Rob Koeling, J. Weeds, and J. Carroll. Using Automatically Acquired Predominant Senses for Word Sense Disambiguation. In: *Senseval-3: Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*. Ed. by Rada Mihalcea and Phil Edmonds. Barcelona, Spain: Association for Computational Linguistics, July 2004, pp. 151–154.

McCarty, Willard. Humanities Computing. In: *Encyclopedia of Library and Information Science*. Ed. by Miriam Drake. 2nd ed. New York: Marcel Dekker, Inc., 2003, pp. 1224–1235.

McKeown, Kathleen and Dragomir R. Radev. Generating summaries of multiple news articles. In: *SIGIR '95: Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. Ed. by Edward A. Fox, Peter Ingwersen, and Raya Fidel. Seattle, Washington, USA: ACM, July 1995, pp. 74–82.

Merriam-Webster Dictionary. *Myth*. URL: http://www.merriam-webster.com/dictionary/myth (visited on 02/10/2014).

Michaels, Axel. The Grammar of Rituals. In: *Grammars and Morphologies of Ritual Practices in Asia*. Ed. by Axel Michaels and Anand Mishra. Vol. 1. Ritual Dynamics and the Science of Ritual. Harrassowitz, Wiesbaden, Dec. 2010, pp. 7–28.

– A Preliminary Grammar of Newar Life-Cycle Rituals. In: *The Journal of Hindu Studies* 5 (1), 2012, pp. 10–29.

Moretti, Franco. Conjectures on World Literature. In: *New Left Review* 1, 2000, pp. 54–68.

Navigli, Roberto. Word Sense Disambiguation: A Survey. In: *ACM Computing Surveys* 41 (2), Feb. 2009.

Navigli, Roberto and Paola Velardi. Automatic Adaptation of WordNet to Domains. In: *Proceedings of workshop OntoLex'2 Ontologies and Lexical Knowledge Bases*. Ed. by Kiril Simov. Las Palmas, Spain: European Language Resources Association (ELRA), May 2002, pp. 1023–1027.

Needleman, Saul B. and Christian D. Wunsch. A general method applicable to the search for similarities in the amino acid sequence of two proteins. In: *Journal of Molecular Biology* 48 (3), Mar. 1970, pp. 443–453.

Ng, Hwee Tou and Hian Beng Lee. Integrating Multiple Knowledge Sources to Disambiguate Word Sense: An Exemplar-Based Approach. In: *Proceedings of the 34th Annual Meeting of the Association for Computational Linguistics*. Santa Cruz, California, USA: Association for Computational Linguistics, June 1996, pp. 40–47.

Niles, Ian and Adam Pease. Towards a Standard Upper Ontology. In: *FOIS '01: Proceedings of the International Conference on Formal Ontology in Information Systems*. Ogunquit, Maine: ACM, Oct. 2001.

Och, Franz Josef and Hermann Ney. A Systematic Comparison of Various Statistical Alignment Models. In: *Computational Linguistics* 29 (1), Mar. 2003, pp. 19–51.

Oppitz, Michael. Montageplan von Ritualen. In: *Rituale heute; Theorien – Kontroversen – Entwürfe*. Ed. by Corina Caduff and Joanna Pfaff-Czarnecka. Berlin: Reimer, 1999, pp. 73–99.

Parker, Robert, David Graff, Junbo Kong, Ke Chen, and Kazuaki Maeda. *English Gigaword Fifth Edition*. Linguistic Data Consortium, Philadelphia. 2011. URL: http://catalog.ldc.upenn.edu/LDC2011T07 (visited on 01/14/2014).

Pedersen, Ted. *WordNet::Similarity*. 2014. URL: http://wn-similarity.sourceforge.net (visited on 02/10/2014).

Plank, Barbara. Structural Correspondence Learning for Parse Disambiguation. In: *Proceedings of the Student Research Workshop at EACL 2009*. Ed. by Vera Demberg, Yanjun Ma, and Nils Reiter. Athens, Greece: Association for Computational Linguistics, Apr. 2009, pp. 37–45.

Presner, Todd and Chris Johanson. *The Promise of Digital Humanities*. Whitepaper. 2009. URL: http://humanitiesblast.com/Promise%20of%20Digital%20Humanities.pdf (visited on 01/14/2014).

Propp, Vladimir Yakovlevich. *Morphology of the Folktale*. 2nd. Translated by Laurence Scott (Original work published 1928). Austin, TX: University of Texas Press, 1958.

Raben, Joseph. Humanities Computing 25 Years Later. In: *Computers and the Humanities* 25, 1991, pp. 341–350.

Rand, William M. Objective Criteria for the Evaluation of Clustering Methods. English. In: *Journal of the American Statistical Association* 66 (336), Dec. 1971, pp. 846–850.

Rayson, Paul and Roger Garside. Comparing Corpora using Frequency Profiling. In: *The Workshop on Comparing Corpora*. Ed. by Adam Kilgarriff and Tony Berber Sardinha. Hong Kong, China: Association for Computational Linguistics, Oct. 2000, pp. 1–6.

Recasens, Marta and Eduard Hovy. BLANC: Implementing the Rand index for coreference evaluation. In: *Natural Language Engineering* 17, 04 Sept. 2011, pp. 485–510.

Reddy, Siva, Abhilash Inumella, Diana McCarthy, and Mark Stevenson. IIITH: Domain Specific Word Sense Disambiguation. In: *Proceedings of the 5th International Workshop on Semantic Evaluation*. Ed. by Katrin Erk and Carlo Strapparava. Association for Computational Linguistics. Uppsala, Sweden, July 2010, pp. 387–391.

Regneri, Michaela, Alexander Koller, and Manfred Pinkal. Learning script knowledge with web experiments. In: *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*. Ed. by Jan Hajič, Sandra Carberry, Stephen Clark, and Joakim Nivre. Uppsala, Sweden: Association for Computational Linguistics, July 2010, pp. 979–988.

Reiter, Nils and Anette Frank. Identifying Generic Noun Phrases. In: *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*. Ed. by Jan Hajič, Sandra Carberry, Stephen Clark, and Joakim Nivre. Uppsala, Sweden: Association for Computational Linguistics, July 2010, pp. 40–49.

Reiter, Nils, Oliver Hellwig, Anette Frank, Irina Gossmann, Borayin Maitreya Larios, Julio Rodrigues, and Britta Zeller. Adapting NLP Tools and Frame-Semantic Resources for the Semantic Analysis of Ritual Descriptions. In: *Language Technology for Cultural Heritage: Selected Papers from the LaTeCH Workshop Series*. Ed. by Caroline Sporleder, Antal van den Bosch, and Kalliopi A. Zervanou. Theory and Applications of Natural Language Processing. Berlin, Heidelberg: Springer, 2011, pp. 171–193.

Riloff, Ellen and William Phillips. *An Introduction to the Sundance and AutoSlog Systems*. Tech. rep. UUCS-04-015. School of Computing, University of Utah, 2004.

Roberts, Brian. *Biographical Research*. Understanding Social Research. Buckingham, Philadelphia: Open University Press, 2002.

Rocchio, Joseph John. Relevance feedback in information retrieval. In: *The SMART Retrieval System: Experiments in Automatic Document Processing*. Ed. by Gerard Salton. Prentice-Hall Series in Automatic Computation. Englewood Cliffs NJ: Prentice-Hall, 1971. Chap. 14, pp. 313–323.

Roth, Michael. Inducing Implicit Arguments via Cross-document Alignment – A Framework and its Applications. Defended on December 3rd, 2013. PhD thesis. Heidelberg University, 2014.

Roth, Michael and Anette Frank. Aligning Predicates across Monolingual Comparable Texts using Graph-based Clustering. In: *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*. Ed. by Jun'ichi Tsujii, James Henderson, and Marius Paşca. Jeju Island, Korea: Association for Computational Linguistics, July 2012, pp. 171–182.

Rus, Vasile, Mihai Lintean, Rajendra Banjade, Nobal Niraula, and Dan Stefanescu. SEMILAR: The Semantic Similarity Toolkit. In: *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics: System Demonstrations*. Ed. by Miriam Butt and Sarmad Hussain. Sofia, Bulgaria: Association for Computational Linguistics, Aug. 2013.

Sagae, Kenji and Jun'ichi Tsujii. Dependency Parsing and Domain Adaptation with LR Models and Parser Ensembles. In: *Proceedings of the CoNLL Shared Task Session of EMNLP-CoNLL 2007*. Ed. by Jason Eisner. Prague, Czech Republic: Association for Computational Linguistics, June 2007, pp. 1044–1050.

Sandu, Oana, Giuseppe Carenini, Gabriel Murray, and Raymond Ng. Domain Adaptation to Summarize Human Conversations. In: *Proceedings of the 2010 Workshop on Domain Adaptation for Natural Language Processing*. Ed. by Hal Daumé III, Tejaswini Deoskar, David McClosky, Barbara Plank, and Jörg Tiedemann. Uppsala, Sweden: Association for Computational Linguistics, July 2010, pp. 16–22.

Sang, Erik F. Tjong Kim and Sabine Buchholz. Introduction to the CoNLL-2000 Shared Task: Chunking. In: *Proceedings of Fourth Conference on Computational Natural Language Learning and of the Second Learning Language in Logic Workshop*. Lisbon, Portugal: Association for Computational Linguistics, Sept. 2000.

Sculley, D. and Bradley M. Pasanek. Meaning and mining: the impact of implicit assumptions in data mining for the humanities. In: *Literary and Linguistic Computing* 23 (4), 2008, pp. 409–424.

Shimizu, Nobuyuki and Hiroshi Nakagawa. Structural Correspondence Learning for Dependency Parsing. In: *Proceedings of the CoNLL Shared Task Session of EMNLP-CoNLL 2007*. Ed. by Jason Eisner. Prague, Czech Republic: Association for Computational Linguistics, June 2007, pp. 1166–1169.

Soon, Wee Meng, Daniel Chung Yong Lim, and Hwee Tou Ng. A Machine Learning Approach to Coreference Resolution of Noun Phrases. In: *Computational Linguistics* 27 (4), Dec. 2001, pp. 521–544.

Staal, Frits. *Rules without Meaning. Ritual, Mantras and the Human Sciences*. Vol. 4. Toronto Studies in Religion. New York: Peter Lang, 1989.

Stanford NLP Group. *Frequently Asked Questions*. 2014. URL: `http://nlp.stanford.edu/software/parser-faq.shtml%7B%5C#%7Dz` (visited on 02/10/2014).

Stevenson, Mark, Eneko Agirre, and Aitor Soroa. Exploiting domain information for Word Sense Disambiguation of medical documents. In: *Journal of the American Medical Informatics Association* 19 (2), 2012, pp. 235–240.

Stolcke, Andreas and Stephen Omohundro. Hidden Markov Model Induction by Bayesian Model Merging. In: *Advances in Neural Information Processing Systems*. Ed. by Steve J. Hanson, J. D. Jack D. Cowan, and C. Lee Giles. Vol. 5. San Mateo, California: Morgan Kaufmann, 1993, pp. 11–18.

Tanca, Antonio. *Foreign armed intervention in internal conflict*. Dordrecht: Martinus Nijhoff, 1993.

Tiedemann, Jörg. Recycling Translations – Extraction of Lexical Data from Parallel Corpora and their Application in Natural Language Processing. Anna Sågvall Hein, Åke

Viberg (eds): Studia Linguistica Upsaliensia. PhD thesis. Uppsala, Sweden: Uppsala University, 2003.

Uther, Hans-Jörg. *The Types of International Folktales: A Classification and Bibliography. Based on the system of Antti Aarne and Stith Thompson.* FF Communications 284–286. Helsinki: Suomalainen Tiedeakatemia, 2004.

Velardi, Paola, Michele Missikoff, and Roberto Basili. Identification of relevant terms to support the construction of Domain Ontologies. In: *Proceedings of the ACL 2001 Workshop on Human Language Technology and Knowledge Management*. Ed. by Mark Maybury, Niels Ole Bernsen, and Steven Krauwer. Toulouse, France: Association for Computational Linguistics, July 2001.

Versley, Yannick, Simone Paolo Ponzetto, Massimo Poesio, Vladimir Eidelman, Alan Jern, Jason Smith, Xiaofeng Yang, and Alessandro Moschitti. BART: A Modular Toolkit for Coreference Resolution. In: *Proceedings of the ACL-08: HLT Demo Session*. Ed. by Johanna D. Moore, Simone Teufel, James Allan, and Sadaoki Furui. Columbus, Ohio: Association for Computational Linguistics, June 2008, pp. 9–12.

Vilain, Marc, John Burger, John Aberdeen, Dennis Connolly, and Lynette Hirschman. A Model-Theoretic Coreference Scoring Scheme. In: *Sixth Message Understanding Conference (MUC-6): Proceedings of a Conference Held in Columbia*. Columbia, Maryland, Nov. 1995.

Voormann, Holger and Ulrike Gut. Agile corpus creation. In: *Corpus Linguistics and Linguistic Theory* 4 (2), Dec. 2008, pp. 235–251.

Yimam, Seid Muhie, Iryna Gurevych, Richard Eckart de Castilho, and Chris Biemann. WebAnno: A Flexible, Web-based and Visually Supported System for Distributed Annotations. In: *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics: System Demonstrations*. Ed. by Miriam Butt and Sarmad Hussain. Sofia, Bulgaria: Association for Computational Linguistics, Aug. 2013, pp. 1–6.