EuroVis Workshop on Visual Analytics (2017)
M. Sedlmair and C. Tominski (Editors)

# Feature Alignment for the Analysis of Verbatim Text Transcripts

Wolfgang Jentner, Mennatallah El-Assady, Bela Gipp, and Daniel Keim

University of Konstanz, Germany

**Abstract**

*In the research of deliberative democracy, political scientists are interested in analyzing the communication models of discussions, debates, and mediation processes with the goal of extracting reoccurring discourse patterns from the verbatim transcripts of these conversations. To enhance the time-exhaustive manual analysis of such patterns, we introduce a visual analytics approach that enables the exploration and analysis of repetitive feature patterns over parallel text corpora using feature alignment. Our approach is tailored to the requirements of our domain experts. In this paper, we discuss our visual design and workflow, and we showcase the applicability of our approach using an experimental parallel corpus of political debates.*

## 1. Introduction

Political debates and mediations are widely studied across the humanities and social sciences due to their often long-lasting political and social impact. The verbatim transcripts of these conversations capture the rapid exchange of opinions and arguments between speakers. Political scientists and linguists are interested in the relations of automatically generated text features [GEH*17] shedding light on (un-)intentionally used tactics, strategies, and common patterns within debates and discussions. This may include certain verbal actions triggering reactions or common frequent sequences in discussions emphasized by specific text features. However, it is currently a time-extensive task to verify theoretically-motivated hypotheses with empirical results. In a typical analysis process, domain experts rely on statistical tests to aid their manual examination of simple discourse patterns over parallel corpora.

The main contribution of our approach is that it fills this methodological gap in political science research by enabling domain experts to explore and analyze repetitive patterns over parallel text corpora using automatic text-feature alignment. Our approach embeds sequential pattern mining (SPM) techniques into a visual analytics platform that incorporates user feedback and domain knowledge. Arbitrary text features provided by the VisArgue framework are transformed into discrete events suitable for the SPM algorithm. The mined sequential patterns incorporate gaps and allow for incomplete matches. The visualization is steered by the users to highlight relevant patterns across different conversations, as well as, repetitive patterns within the same document. Our approach supports the understanding of abstract patterns in conversations through interactive close- and distant-reading views. With this visual analytics approach, we enable political science researchers to generate and verify hypotheses about their data, going beyond manual statistical testing and correlation analysis. Through the close-reading components, the domain experts are additionally supported in understanding and explaining the found patterns.

Domain experts in political science currently apply time-exhaustive, manual analysis, in combination with basic statistical processing to find correlations and simple patterns in the data. Together with domain experts, we derived the following requirements: **[R1]** Extracting patterns combining multiple abstract text features, e.g. topics, speakers, sentiment. Patterns may contain gaps and must not match completely. **[R2]** Finding *repetitive patterns* independently in *one* sequence. The patterns should have the same characteristics as in [R1]. **[R3]** Interactive pattern generation and search (hypothesis generation). **[R4]** Tight integration of close- and distant-reading (hypothesis verification). **[R5]** Explorative analysis should allow $n : n$ document-comparisons. **[R6]** Visualization of patterns and their alignments (including $n : n$ comparisons).

## 2. Related Work

Existing visualizations, algorithms and interactive applications cover some of the specified requirements. To the best of our knowledge, no system exists that is able to fulfill all of the requirements. The used data consists of long and dense event sequences that make a visual identification of related events impossible. To assist the user with this task, we use a sequential pattern mining (SPM) algorithm to detect patterns automatically. The problem of SPM was first formalized by Agrawal et al. [AS95] and flourished in the development of numerous algorithms [ME10]. A vertical database in combination with bitmap representation [AFGY02, AOV06] has shown to be very efficient [ME10, FGCT14] in SPM.

Text alignment applications are most closely related to our approach and are typically used in fields such as machine translation [GC93], text simplification [BS11], speech analysis [HK07], or plagiarism detection [Gip14, OHR11]. Such tools work on single event sequences and align two documents. In some cases, a visualization is provided, which displays these alignments to the user. In general, such approaches are primarily designed to calculate similarity scores. Lent et al. [LAS97] use sequential pattern mining to



DOI: 10.2312/eurova.20171113

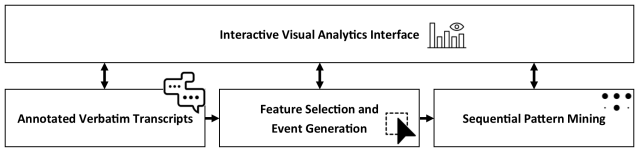



**Figure 1:** *The users follows a workflow starting with annotated verbatim transcripts. The extracted text features are transformed to events to be suitable for a SPM algorithm. All of the steps feature some interactive visualization to steer and refine the parameters and explore the found patterns.*

find trending patterns in text. Sun et al. [SLC*07] use sequential patterns to detect erroneous sentences based on POS tags. Wong et al. [WCF*00] visualize sequential patterns to reveal relations of topics mined from large text corpora. *ViTA* [ARO*17] allows users to define their own models and align two documents based on the models. Sequences can be visually detected by parallel lines connecting the single matches, as well as using dot plot visualization when a series occurs. *PoemVis* [ALC*13] is able to detect repeating phonetic features. Wattenberg uses arc diagrams to visualize repeating sequences in strings [Wat02]. Both tools are not capable of analyzing multiple documents at the same time. The tools allow their users to (visually) find patterns. However, no automatic pattern detection is used, or it only works on single-event sequences and with two documents.

## 3. Approach

The application described in this paper is embedded into the VisArgue framework [EAGJ*16]. The pipeline in Figure 1 shows the workflow. The user can upload and select multiple annotated text transcripts for analysis. Various applications can be selected depending on the analysis task. The approach presented in this paper is one of these applications. The user may modify a range of parameters during preprocessing. In the application presented in this paper, the user proceeds with the event generation step (Section 3.1). A SPM algorithm finds SPs based on the generated events (Section 3.2) and these SPs are then presented to the user and can be filtered in the interactive sequence visualization (Section 3.3).

### 3.1. Event Generation

A SPM algorithm uses discrete events as input, which are characterized by two attributes: (1) an identification, where events with the same id share the same semantics and content, (2) a timestamp to define the order of events. It is possible that two events with different ids occur at the same point in time. In this section, we will detail the required transformation of text features into discrete events for the pattern detection **[R1]**.

The VisArgue framework generates text features primarily based on annotations. A text feature consists of a name (e.g. speaker) and two or more characteristics (e.g. name1, name2). The event id is generated based on the concatenation. We divide the text features into three categories. The user can modify the names of characteristics and merge them in the case of *binary* or *discrete* text features. For *continuous* text features, the user may specify the limits for the bins and name them. To guarantee a fast mining of the patterns, up to 5 text features can be selected. All settings are stored, allowing the user to switch back to this stage and refine her settings.

### 3.2. Sequential Pattern Mining

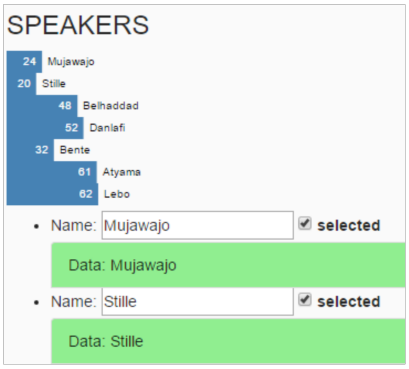
It is difficult to find complex patterns visually. To support the experts, we detect patterns automatically and present them to the user. A SPM algorithm that outputs sequential patterns (SPs) and their positions within the sequences is sought because these patterns fulfill **[R1]**. Our proposed algorithm is inspired by CM-SPAM [FGCT14]. We extend the CMAP to a complete index, storing all positions in all sequences for each tuple satisfying the given minimum support. The pattern generation only scans the generated index and tests whether itemset and sequence extensions are possible. All occurrence positions are stored for each pattern and must be joined after a successful extension. This makes our approach considerably slower than CM-SPAM but still provides fast results in this application and with the current hardware setting. To detect repetitive SPs within one sequence **[R2]**, the support calculation is modified to use the number of occurrences instead of counting the sequences. We take a max-gap constraint [SA96] into consideration, assuming that users are not interested in patterns with large gaps. The frequent SPs are stored in a prefix-tree. We post-process this tree such that it stores all pattern containments in order to precalculate sets for maximal and closed SPs [FWGT14]. We introduce a set of occurrence-closed SPs, which is similar to closed SPs, but also considers the number of occurrences.

After the event generation stage, the user can define the minimum support and the max-gap constraint. The application then proceeds to the interactive sequence visualization stage in which the automatically detected SPs and their alignments are displayed to the user.

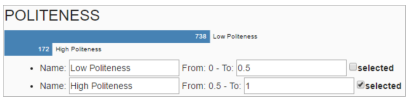
### 3.3. Interactive Sequence Visualization

Initially, the user is presented with the overview sequence visualization (Figure 2c). The user set the minimum support to 100% and the max-gap constraint to 1 which allows no gaps in the patterns.

**Sequences** A colored circle represents an event. Each event has its own color assigned, provided by the HSLuv color space [Bor16]. The colors can be modified by the user. Hovering over an event will show a tooltip revealing the name of the text feature and its characteristic (Section 3.1). Additionally, all events with the same id are highlighted in the sequence by lowering the opacity for the rest of the events. The circles are vertically grouped to visually distinguish the discussions. In the scenario shown in Figure 2c, four discussions were loaded **[R5]**. Each discussion is labeled by the rotated text on the left showing the filename. Each text feature is represented as a horizontal line of circles and labeled by the feature-name on the left side. The user selected three text features for this task: *politeness*, *speaker*, and *topic*. The vertical order of the text features for each discussion is consistent. Both, the order of the sequences and the order of the text features can be modified by the user. One column of circles represents one utterance in the discussion. It is evident that the first sequence is the shortest. As characteristics can be individually deselected by the user, there might be gaps where no event is present. This visualization is compact but




**(a)** *The histogram and two characteristics for a categorical text feature. The user can move the green bars to merge characteristics.*




**(b)** *Bins and a histogram for numerical text features that can be modified by the user.*


**(c)** *In this scenario, four discussions were loaded and are represented as blocks of circles. Each text feature is displayed as one horizontal line for each sequence. The side bar on the right features a list of detected SPs where the first one is currently hovered and, thus, highlighted in the sequence visualization. Patterns and their positions are connected by grey lines (red if highlighted). The position of patterns that are repeated in one sequence are indicated by the grey horizontal lines above or below each sequence.*

**Figure 2:** *Figures 2a and 2b show configuration options in the event generation step. 2c shows the interactive sequence visualization.*

allows to see all events for each utterance. Furthermore, it is consistent with the pattern visualization.

**Patterns** The list on the top-right side of Figure 2c shows detected patterns. The visualization for one pattern follows the design of a sequence and is inspired by the notation in [WC09]. The text features are listed in the same order as in the sequence. The only difference to the sequence is that patterns will not show gaps as these are abstracted from the original occurrences. Hovering the events in a pattern will also show a tooltip revealing the text feature and characteristic's name. Patterns can be (de-)selected in the list, which will hide their connections in the visualization.

**Alignments** The grey lines indicate the position of a pattern and connect them visually **[R6]**. In Figure 2c, the highlighted pattern establishes that all events that do not belong to this pattern have a lower opacity. Additionally, the stroke-width of the connection is increased and colored red. The same design is used for repetitions. Here, the lines are horizontal and placed above or below each sequence **[R2,R6]**. This ensures a compact representation as opposed to arc diagrams. In cases where the minimum support is less than 100%, a connecting line might skip a sequence, which results in crossing over one or more sequences. This can be minimized when the user manually reorders the sequences.

**Filters** Various filters are provided (bottom Figure 2c). These filters allow the user to search for specific properties of a pattern, including the minimum number of text features, items, and occurrences. Additionally, the user may also select specific text features and characteristics that a pattern must contain. Although none of these filters require a re-computation of the SPM algorithm, the system verifies that the limits and properties are satisfied by a single run through the result set. This allows the user to interactively explore the frequent patterns **[R3]**. However, the user may also always switch back to the event generation step and modify the available settings there.

**Close-Reading** The system enables users to investigate the utterances and patterns in more detail. By clicking one of the grey connections, a pop-up will show the text of two discussions side by side similarly to a diff view (Figure 4) **[R4]**. The top displays the selected pattern. The user may choose two alignment positions at the same time. The colored bars in the center represent the event belonging to the utterance. Their order (left to right) equals the vertical order of the text features in the sequence and pattern respectively and use the same colors as the events. Since some of the text features are based on annotations, the respective parts of the annotation are highlighted in the text. This helps users gain further insight on the text features and helps them understand and verify the patterns. Clicking an event in a sequence opens a similar presentation where only one utterance of one discussion is displayed.

## 4. Expert Study

We conducted an expert study with two political scientists at the Ph.D. student and post-doc level and one linguist at a post-doc level. All of the experts are heavily involved in the VisArgue project. An expertise in this area is required to be able to gain insight from the found SPs. The selected data origins from experiments (*AfricaGames*) conducted by the political scientist participants of our expert study. Study participants of the *AfricaGames* have to take over a role of a fictional African king and discuss a new political system in their fictional country. Each experiment is performed with a different group of five students and one moderator who is the political scientist expert in our study. The discussions were recorded, transcribed, and automatically annotated.

The expert study was conducted in single sessions, the screen, and the expert's voices were recorded. The experts were asked to speak out loud their ideas, insights, actions, and experiences. After a short introduction of the tool, the task was to formulate hypotheses and expectations towards the application. Afterward, the experts tried to verify or reject their previously stated hypotheses and explored the data and its patterns in more detail. In the end, the experts gave a summary with feedback about their workflow, their success, and their own subjective efficiency. The experts used the previously described data from the controlled experiments. Eight files were available and selected by the experts in the preprocessing stage. No time constraint is set and the length of the sessions varied between one to three hours.

Expectations ranged from finding similarities in discussions, understanding the process of decision making, seeing an increase of patterns containing the feature *agreement* towards the end of each discussion, finding co-occurring text features, and text features triggering other text features.

The experts started with simple hypotheses typically referring to known co-occurring text features and simple patterns (e.g *condition* and *consequence*). The experts had no problems using the tool and quickly started to test more complex hypotheses and to explore the data. All three experts find the highlighted pattern in Figure 2c. This is expected since this pattern occurs due to an introduction round of the participants in the *AfricaGames*. The participants always introduced themselves in the same order. However, the experts are surprised by the prominence of the pattern at a minimum support of 100% as not only the speaker events are matching but also topics and some other text features. A typical showcase is displayed in Figure 3, where the expert looks for variations of the text features *bargaining* and *consensus* because it is expected that phases of bargaining are followed by phases of consensus within one discussion. As such patterns were found, the focus shifted towards the positions and the number of occurrences in the discussions, followed by a close-reading phase (Figure 4) in which the annotations were investigated in more detail. If the experts could not find the desired patterns due to very strict constraint settings, a typical strategy was to lower the minimum support and/or increase the max-gap parameter. This relaxation of constraints was performed multiple times until the pattern or a similar one could be found or the expert rejected the hypothesis. The experts felt very comfortable in using the tool, a highly appreciated feature was the text-diff-view, since detected patterns could be instantly investigated. During the study, multiple unexpected patterns were identified, where it was unclear

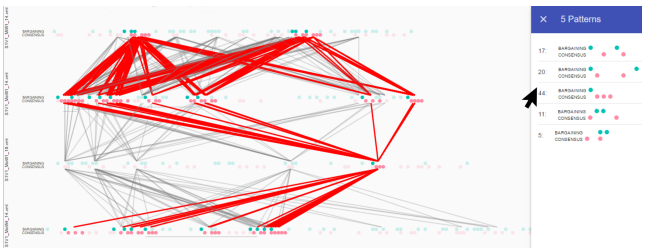



**Figure 3:** *The screenshot shows different patterns where bargaining and consensus are alternating. One pattern is currently highlighted.*

**Figure 4:** *The diff view allows a close-reading and side by side comparison of utterances belonging to selected patterns, e.g. Fig. 3.*

if these occurred randomly or had a deeper meaning. The experts mentioned that such patterns served as starting points for new research in their fields.

## 5. Discussion and Conclusions

The application presented in this paper enables its users to find, explore, and verify frequent sequential patterns based on various text features in multiple documents and within the same document. The event generation step serves as an abstraction and allows the analyst to add more text features in the future.
Visualizing the events of a sequence is not sufficient to detect complex patterns. Therefore, a SPM algorithm is used to find SPs automatically. The experts had no problems interpreting the patterns and using the application, however, domain knowledge *is* required. The application is limited in its scalability, in that very long sequences with many identical events increase the runtime, since many combinations and positions of occurrence must be calculated. This is also true for a low minimum support or a high max-gap constraint. However, this can be steered by the user and did not occur during the expert study. The number of discussions/sequences have a smaller impact when a high minimum support is maintained. However, the space for the visualization is limited. Not all of the sequences can be displayed at the same time on the screen. The scalability of long sequences is also visually limited. A semantic zoom collapsing the single features into smaller blocks and decreasing the space between the sequences could help overcome both issues to a certain extent. Sparse sequences can be compressed if the user is not interested in absolute occurrence positions. In some exploration phases, a lot of patterns are retrieved. It would be desirable to filter for relevant patterns according to expert judgment. However, since this type of exploration is also new for the experts, a quantification for the relevance of such patterns could not be defined and it is likely task dependent. We could observe that, in general, the experts were more interested in less complex patterns that occurred multiple times in the discussions than complex patterns that only occurred at the specified minimum support. Although of the specific requirements, this application can be used in different areas, for example, plagiarism detection based on citations and other features serving as events. Despite the text domain, further application areas are imaginable such as audio analysis, video analysis, or time series analysis with a modified event generation step. Many tasks require an inspection of found patterns for their validation and verification.

## Acknowledgments

This work was supported by the EU project VALCRI under grant number FP7-SEC-2013-608142.

## References


[AFGY02] AYRES J., FLANNICK J., GEHRKE J., YIU T.: Sequential pattern mining using a bitmap representation. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, July 23-26, 2002, Edmonton, Alberta, Canada* (2002), pp. 429–435. doi:10.1145/775047.775109. 1

[ALC*13] ABDUL-RAHMAN A., LEIN J., COLES K., MAGUIRE E., MEYER M. D., WYNNE M., JOHNSON C. R., TREFETHEN A. E., CHEN M.: Rule-based visual mappings - with a case study on poetry visualization. *Comput. Graph. Forum 32*, 3 (2013), 381–390. doi:10.1111/cgf.12125. 2

[AOV06] ASEERVATHAM S., OSMANI A., VIENNET E.: bitspade: A lattice-based sequential pattern mining algorithm using bitmap representation. In *Proceedings of the 6th IEEE International Conference on Data Mining (ICDM 2006), 18-22 December 2006, Hong Kong, China* (2006), pp. 792–797. doi:10.1109/ICDM.2006.28. 1

[ARO*17] ABDUL-RAHMAN A., ROE G., OLSEN M., GLADSTONE C., WHALING R., CRONK N., MORRISSEY R., CHEN M.: Constructive visual analytics for text similarity detection. *Comput. Graph. Forum 36*, 1 (2017), 237–248. doi:10.1111/cgf.12798. 2

[AS95] AGRAWAL R., SRIKANT R.: Mining sequential patterns. In *Proceedings of the Eleventh International Conference on Data Engineering, March 6-10, 1995, Taipei, Taiwan* (1995), pp. 3–14. doi:10.1109/ICDE.1995.380415. 1

[Bor16] BORONINE A.: Hsluv - human friendly hsl, 2016. URL: http://www.hsluv.org/. 2

[BS11] BOTT S., SAGGION H.: An unsupervised alignment algorithm for text simplification corpus construction. In *Proceedings of the Workshop on Monolingual Text-To-Text Generation* (2011), Association for Computational Linguistics, pp. 20–26. 1

[EAGJ*16] EL-ASSADY M., GOLD V., JENTNER W., BUTT M., HOLZINGER K., KEIM D. A.: VisArgue - A Visual Text Analytics Framework for the Study of Deliberative Communication. In *Proceedings of The International Conference on the Advances in Computational Analysis of Political Text (PolText2016)* (Zagreb, 2016), University of Zagreb, pp. 31–36. URL: http://visargue.inf.uni-konstanz.de/. 2

[FGCT14] FOURNIER-VIGER P., GOMARIZ A., CAMPOS M., THOMAS R.: Fast vertical mining of sequential patterns using co-occurrence information. In *Advances in Knowledge Discovery and Data Mining - 18th Pacific-Asia Conference, PAKDD 2014, Tainan, Taiwan, May 13-16, 2014. Proceedings, Part I* (2014), pp. 40–52. doi:10.1007/978-3-319-06608-0_4. 1, 2

[FWGT14] FOURNIER-VIGER P., WU C., GOMARIZ A., TSENG V. S.: VMSP: efficient vertical mining of maximal sequential patterns. In *Advances in Artificial Intelligence - 27th Canadian Conference on Artificial Intelligence, Canadian AI 2014, Montréal, QC, Canada, May 6-9, 2014. Proceedings* (2014), pp. 83–94. doi:10.1007/978-3-319-06483-3_8. 2

[GC93] GALE W. A., CHURCH K. W.: A program for aligning sentences in bilingual corpora. *Computational Linguistics 19*, 1 (1993), 75–102. 1

[GEH*17] GOLD V., EL-ASSADY M., HAUTLI-JANISZ A., BÖGEL T., ROHRDANTZ C., BUTT M., HOLZINGER K., KEIM D. A.: Visual linguistic analysis of political discussions: Measuring deliberative quality. *DSH 32*, 1 (2017), 141–158. doi:10.1093/llc/fqv033. 1

[Gip14] GIPP B.: *Citation-based Plagiarism Detection - Detecting Disguised and Cross-language Plagiarism using Citation Pattern Analysis*. Springer, 2014. URL: http://dx.doi.org/10.1007/978-3-658-06394-8, doi:10.1007/978-3-658-06394-8. 1

[HK07] HAUBOLD A., KENDER J. R.: Alignment of speech to highly imperfect text transcriptions. In *Proceedings of the 2007 IEEE International Conference on Multimedia and Expo, ICME 2007, July 2-5, 2007, Beijing, China* (2007), pp. 224–227. doi:10.1109/ICME.2007.4284627. 1

[LAS97] LENT B., AGRAWAL R., SRIKANT R.: Discovering trends in text databases. In *Proceedings of the Third International Conference on Knowledge Discovery and Data Mining (KDD-97), Newport Beach, California, USA, August 14-17, 1997* (1997), pp. 227–230. 1

[ME10] MABROUKEH N. R., EZEIFE C. I.: A taxonomy of sequential pattern mining algorithms. *ACM Comput. Surv. 43*, 1 (2010), 3:1–3:41. doi:10.1145/1824795.1824798. 1

[OHR11] OLSEN M., HORTON R., ROE G.: Something borrowed: sequence alignment and the identification of similar passages in large text collections. *Digital Studies/Le champ numérique 2*, 1 (2011). 1

[SA96] SRIKANT R., AGRAWAL R.: Mining sequential patterns: Generalizations and performance improvements. In *Advances in Database Technology - EDBT'96, 5th International Conference on Extending Database Technology, Avignon, France, March 25-29, 1996, Proceedings* (1996), pp. 3–17. doi:10.1007/BFb0014140. 2

[SLC*07] SUN G., LIU X., CONG G., ZHOU M., XIONG Z., LEE J., LIN C.: Detecting erroneous sentences using automatically mined sequential patterns. In *ACL 2007, Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics, June 23-30, 2007, Prague, Czech Republic* (2007). 2

[Wat02] WATTENBERG M.: Arc diagrams: Visualizing structure in strings. In *2002 IEEE Symposium on Information Visualization (InfoVis 2002), 27 October - 1 November 2002, Boston, MA, USA* (2002), pp. 110–116. doi:10.1109/INFVIS.2002.1173155. 2

[WC09] WU S., CHEN Y.: Discovering hybrid temporal patterns from sequences consisting of point- and interval-based events. *Data Knowl. Eng. 68*, 11 (2009), 1309–1330. doi:10.1016/j.datak.2009.06.010. 3

[WCF*00] WONG P. C., COWLEY W., FOOTE H., JURRUS E., THOMAS J.: Visualizing sequential patterns for text mining. In *IEEE Symposium on Information Visualization 2000 (INFOVIS'00), Salt Lake City, Utah, USA, October 9-10, 2000.* (2000), pp. 105–111. doi:10.1109/INFVIS.2000.885097. 2