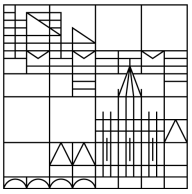
MASTER THESIS

AUTOMATIC AND VISUAL THREAD-RECONSTRUCTION OF CONVERSATIONAL TEXT

RITA SEVASTJANOVA

Universität Konstanz



1.EVALUATED BY Prof. Dr. Daniel Keim
2.EVALUATED BY Junior-Prof. Dr. Bela Gipp

Department of Computer and Information Science
Data Analysis and Visualization Group
University of Konstanz

September 2017

SUPERVISOR:
Prof. Dr. Daniel Keim
Junior-Prof. Dr. Bela Gipp

ADVISOR:
Mennatallah El-Assady

LOCATION:
Konstanz, Germany

ABSTRACT

Online environments, such as forums, encode a lot of knowledge about people's viewpoints on different issues. As an indicator of stances and opinions in society, it is crucial to gather this data in order to structure and analyze its content. However, due to their *implicit conversational structure*, information contained in these mediums is not readily available for analysis. Furthermore, not many forums maintain the logical reply-structure as a publicly accessible interface. Most commonly, only a temporally-ordered sequence of messages within a thread is provided for usage and further analysis. Hence, to observe some existing patterns in the data, scholars rely on automatic techniques to reconstruct the reply-relation structure.

Despite the prior research where different (unsupervised and supervised) methods are used to rebuild the thread structure, some challenges remain open. The most important challenges are:

- **Restricted Applicability:** Most of the supervised models are created to fit one specific dataset.
- **Short Discussions:** Many existing models perform well on short discussions; the performance decreases with increasing message count.
- **Insufficient Evaluation:** Usually, the model's performance is described using evaluation metrics, such as precision and recall. Sometimes, statistical results alone may be misleading (e.g., if the model has overfitted the training data).

To address all previously mentioned challenges, we present an automatic and visual approach to reconstruct the thread-structure of conversational text data. To the best of our knowledge, no visual analytics approach exists which can be used for the reply-relation reconstruction task.

In order to ensure a **broader applicability**, the approach provides both, a supervised classifier and an unsupervised query based model, to reconstruct the reply-relation structure. For the latter one, altogether 17 features can be combined to fit the input discussion best. This is the first contribution of this thesis.

The basis of the second contribution is in the observation that the reconstruction of short discussions is less challenging than of the long ones. Not all messages in a discussion have the same value. Thus, our approach supports a **reduction** of the original discussion to the **most valuable message subset**. Frequently, the reconstruction of this subset is more effective than of the original full discussion.

The third main contribution is the visual representation of the recomputed reply-relation structure, which, besides the statistical evaluation metrics, provides an **evidence on the extracted structure's certainty**. Hence, the system can be used to reconstruct the reply-structure of unseen conversations, or to evaluate different models to each other, having the ground truth information.

ZUSAMMENFASSUNG

Online-Platformen, wie zum Beispiel Online-Foren, enthalten umfangreiches Wissen zu verschiedenen Themen und zu den Meinungen der Teilnehmenden. Zur Analyse dieser Daten ist die zentrale Sammlung der Daten notwendig. Ohne vorherige Bearbeitung/Anreicherung dieser Daten können keine relevanten Informationen, die in diesen Medien enthalten sind, generiert werden. Dies liegt an der impliziten Dikussionsstruktur. Viele Online-Foren stellen diese implizite Struktur nicht zur Verfügung. Häufig enthalten öffentlich zugängliche Daten nur Informationen zur zeitlichen Abfolge der Beiträge. Um neue Muster in diesen Daten erkennen zu können, wendet man zur Rekonstruktion der impliziten Struktur in der Regel zuerst automatisierte Methoden an.

Bereits bestehende, verwandte Arbeiten verwenden zur Rekontruktion von Strukturen sowohl nicht überwachte (*unsupervised*) als auch überwachte (*supervised*) Methoden. Es bestehen jedoch weiterhin komplexe Herausforderungen. Die wesentlichsten Herausforderungen sind:

- **Eingeschränkte Anwendbarkeit:** Die Mehrheit der existierenden Modelle sind nur auf einen bestimmten Datensatz anwendbar.
- **Kurze Diskussionen:** Die Mehrheit der existierenden Modelle liefern lediglich für kurze Diskussionen eine gute Performanz.
- **Unzureichende Evaluierung:** Die Leistung der einzelnen Modelle wird häufig mit Metriken wie *precision* oder *recall* beschrieben. Unter Umständen können solche statistische Merkmale jedoch irreführend sein, zum Beispiel, wenn das Modell aufgrund zu vieler und/oder irrelevanter Trainingsdaten überangepasst ist (*overfitting*).

Um alle gerade erwähnten Herausforderungen angehen zu können, präsentieren wir einen automatisierten, visuellen Ansatz, der die implizite Diskussionsstruktur rekonstruiert. Ein ähnlicher Visual-Analytics Ansatz, der die Diskussionsstruktur wiederherstellt, existiert nach unserem Kenntnisstand zum heutigen Zeitpunkt nicht.

Um eine **breitere Anwendbarkeit** gewährleisten zu können, bietet unser Ansatz sowohl ein überwachtes (*supervised*) als auch ein nicht überwachtes (*unsupervised*), auf Abfragen basierendes Klassifizierungsmodell an. Für das Letztere können insgesamt 17 Merkmale kombiniert werden, die die Eingabedaten am besten räpresentieren. Dies stellt den ersten Forschungsbeitrag der vorliegenden Arbeit dar.

Grundlage des zweiten Forschungsbeitrages ist die Beobachtung darüber, dass kürzere Diskussionen leichter zu rekonstruieren sind als lange Diskussionen. Die einzelnen Nachrichten in einer Diskussion können unterschiedliche Qualität haben. Unser Ansatz erlaubt den Nutzern die Diskussion auf **die wertvollste Teilmenge der Nachrichten** zu verkürzen. Die Wiederherstellung der Struktur einer Teilmenge ist häufig effektiver als die einer Diskussion als Ganzes.

Der dritte Forschungsbeitrag ist die visuelle Darstellung der wiederhergestellten Diskussionsstruktur. Diese Darstellung **zeigt die Bestimmtheit der Struktur** an. Infolgedessen, kann unser System sowohl auf unbekannte Daten, die keine Struktur beinhalten, angewandt werden, als auch auf Daten, die die implizite Struktur beinhalten. Das Letztere erlaubt verschiedliche Modelle miteinander zu vergleichen, um das leistungsfähigste Modell zu bestimmen.

ACKNOWLEDGMENTS

I would like to express my sincere gratitude to my advisor Prof. Dr. Daniel Keim, for giving me the opportunity to write this thesis, and for his encouragement. His excellent classes have inspired me to work on issues of text processing. I would also like to thank the Junior-Professor Dr. Bela Gipp for his support. I am looking forward for potential cooperation in the future.

I would like to gracefully acknowledge my advisor Mennatallah El-Assady, whose motivation, guidance and valuable ideas helped me to complete this work. Thank you for all those discussions in Skype. I appreciate that you always found time for me, even though you were busy with many other projects. Thank you!

Sintija, thank you so much for finding people who agreed to read my thesis!

And finally, I would like to thank my family and my partner Matthias, who are always there for me. Thank you so much for your support!

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACRONYMS

CRF Conditional Random Fields

IR Information Retrieval

NLP Natural Language Processing

RTS Reconstructing Thread Structure

SMSS Simultaneously Model Semantics and Structure

SVM Support Vector Machine

TF-IDF Term Frequency - Inverse Document Frequency

1 INTRODUCTION

Conversational texts, such as forums, have long been a popular option for web users to communicate with others. With millions of users' contribution, a valuable knowledge has been accumulated on various topics, such as politics, society, science, sports, health, etc. [37].

A discussion forum traditionally has a hierarchical structure; it can contain multiple subforums, each of which may have several topics. Within one topic, each new discussion started is called a thread. A message or a post is the smallest discussion unit. It is an utterance written by a user, containing one or several sentences. The first message in a thread is the discussion's title. Two messages can be linked via a reply-relation (parent-child relation), where the parent message is written earlier than the child, and the child responses to the parent. An example of a parent-child relation is shown in Fig. 1

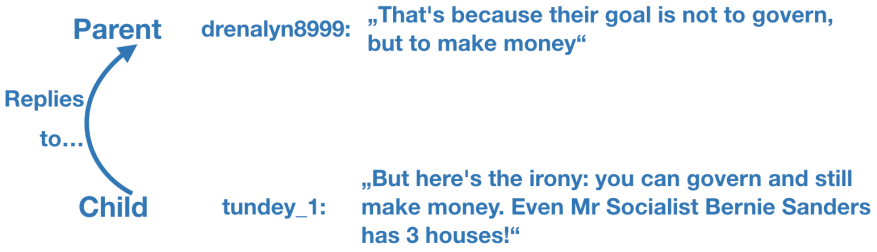


Figure 1: An example of a reply-relation.

Due to their *implicit conversational structure*, information contained in forum discussions is not readily available for analysis. Furthermore, not many forums maintain the logical reply-structure as a publicly accessible interface. Hence, to observe some existing patterns in the data, scholars rely on automatic techniques to reconstruct the reply-relation structure, which is known as *the reconstructing thread structure (RTS)* task [4]. In this thesis, the terms *RTS* and the "reply-relation reconstruction" are used as synonyms.

Conversational text is a challenge for natural language processing (NLP), due to unintentional errors, dialectal variation, conversational ellipsis, topic diversity, and creative use of language and orthography [16]. Messages in threaded discussions are often short, which brings new challenges to the traditional research topics in text analytics, such as text classification, or information extraction [1]. Frequently, such short texts cannot provide sufficient context information for similarity measure, the basis of many text processing methods [45].

To reconstruct the thread structure, one can use supervised or unsupervised methods, both having their pros and cons. Supervised methods, such as various classifi-

cation algorithms, are frequently applied in the prior work, having a higher performance than unsupervised methods. Unsupervised methods are easier to generalize, as no model is learned based on a training data.

Although multiple researchers have taken attempt to deal with the *RTS* task, there are still some major challenges which prohibit us from using one of the existing methods. These challenges are:

- Most of the supervised models are created to fit one specific dataset. Frequently, these models are not applicable on datasets, having different data characteristics.
- Many existing models perform well on short discussions (e.g., threads having ten messages in average). The performance decreases when the model is applied on longer discussions. We see long threads as our target data.
- Usually, the model's performance is described using evaluation metrics, such as precision and recall. Sometimes, a visual representation of computed parent candidates and their certainty level can be more descriptive to outline the model's capability to reconstruct the reply-structure. Such representation could be maintained by a visual analytics tool, but, as far as we know, no visual analytics tool exists which supports the *RTS* task.

We use Reddit[1] forum as our data source. Reddit is an American social news aggregation, web content rating, and discussion website. Reddit's registered community members can submit content such as text posts or direct links. We use this structure as our ground truth, and by applying unsupervised and supervised techniques aim to reconstruct it.

For the reconstruction of the reply-relation structure, we use both - supervised and unsupervised - methods, to compare their performances. Created models are evaluated using a representative sample of 40 Reddit discussions. The characteristics of these files are outlined in Chapter 8.

Machine learning

We begin with a supervised machine learning method. Similarly like in [3], an equally distributed dataset with a similar amount of "positive" and "negative" instances is used to train multiple classifiers (Decision Tree[53] and Random Forest[6] models). The classifiers show a solid performance, by testing them with a 10-fold cross validation technique, but the performance is significantly decreased on the previously mentioned 40 test datasets. Multiple reasons can cause the decrease of the model's performance. Firstly, the identification of reply-relations of relatively long discussions can be seen as *anomaly detection*; the majority of all instances represent the "negative" class. Thus, to deal with this *imbalanced classes problem*, an under-sampling technique is used to generate an equally distributed dataset. By under-sampling the majority class, some relevant information describing the split between two classes ("positive" and "negative") may get lost. Secondly, the performance of the model is highly dependent on the quality of the used training data. If no good split for two target classes exists, it is more likely that the model will overfit the training data.

1 https://www.reddit.com/

Apparently, it is challenging to develop a supervised model which performs well on Reddit forum discussions. Thus, we create an unsupervised query based model. Although the query based model reaches only a modest performance, the performance exceeds that of the machine learning models. The query based model has multiple advantages against the machine learning approach. Firstly, it is more flexible than a trained classifier. A subset of features can be selected to extract the reply-relation structure, describing the input data best. Secondly, it does not overfit. If no reliable features are present in the data, the model can still be used to obtain at least content-full relations, by using the *cosine similarity* function. Thirdly, it is simple to expand the model by adding new features to the feature set.

Query based model

Keim, D.A., Kolhammer J., et al. [32] state that visual analytics "provides technology that combines the strengths of human and electronic data processing." In this semi-automated analytics process, where the capabilities of the machine are combined with those of human, visualizations are used as a medium. These visualizations let the user explore the data from different perspectives and at different levels [32].

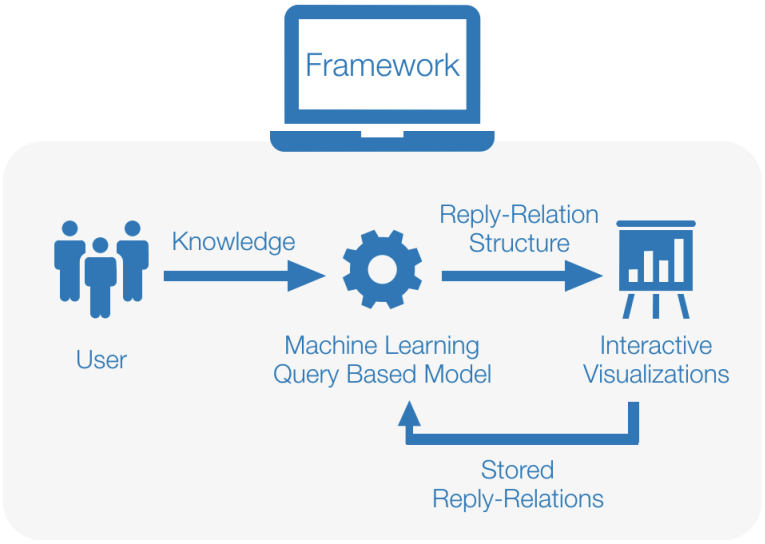


Figure 2: The workflow of the visual analytics tool to reconstruct the reply-relation structure of conversational text data.

We create a visual analytics tool (Fig. 2), which incorporates the users' domain knowledge, and applies automatic methods to reconstruct the reply-structure. The reconstructed structure is displayed in multiple interactive visualizations.

Visual analytics framework

The framework includes the supervised machine learning models and the unsupervised query based model. Currently, the trained classifiers have a poor performance. Thus, the query based model can be applied solely. The tool can be used to explore single model's decisions or compare multiple models to each other. It shows, how certain the extracted relations are. Thus, it is applicable also on a data without the ground truth information. Using different interaction techniques (e.g., *slice and dice*, *brushing*), the user can specify a discussion's subset being of his interest, and work on the reconstruction of this particular subset separately. The intermediate results (reconstructed

reply-relations of this subset) can be stored, allowing an *iterative reconstruction's process*. Such iterative process can help to improve the overall reconstruction's results; different models can be applied on different message subsets.

1.1 REQUIREMENTS

- **R1:** Taking only the temporal and content information of messages into account, the system should reconstruct discussion's reply-relation structure. The model should be flexible, and applicable on various conversational text data.
- **R2:** The quality of messages within a discussion may vary. Thus, sometimes it is sufficient to reconstruct only a subset of present reply-relations, in order to learn discussion's content.
- **R3:** Frequently, the user is not aware how the classification model works. Therefore, the system should provide an evidence which reply-relations may be reliably recreated, and which features influence the classification most.
- **R4:** Discussions about topics being relevant to the society can quickly expand. The system should visually disentangle the complex discussion's structure, and provide insights in its subtopics.

1.2 RESEARCH CHALLENGES

- **RC1:** Regarding the **R1**, the model should be applicable on different conversational text data (e.g., different forum discussions, political discussions). Only the minority of all possible message-pairs in a discussion are correct reply-relations. Hence, it is a challenge to create a good model. Frequently, machine learning methods reach a higher performance than unsupervised methods. But, due to multiple reasons (e.g., a poor quality of data, absence of descriptive features), the supervised models may overfit.
- **RC2:** Regarding the **R2**, the framework should be flexible to integrate the user's domain knowledge. Provided interaction techniques should be simple and intuitive.
- **RC3:** Regarding the **R3**, the system should provide an overview, how the applied features influence the reconstruction's results. The quality of a model should not be described using evaluation metrics alone. Besides the evaluation metrics, the system should give a visual insight, how *close* the reconstructed structure is to the ground truth.
- **RC4:** Regarding the **R4**, visualizations of the threaded discussion should maintain the temporal information of posts, and in the same time disentangle the discussion's reply-structure. They should provide an overview of the whole conversation and show its subtopics on demand.

1.3 CONTRIBUTIONS

- **C1:** To satisfy the **R1** and deal with the **RC1**, we provide a visual analytics tool to reconstruct the reply-relation structure of conversational text data. There, a trained classifier and an unsupervised query based model can be used to deal with the *RTS* task. 17 features are produced to extract the reply-relations, which can be applied on different conversational text data.
- **C2:** Our tool supports an *iterative reconstruction's process*, which satisfies the **RC2**. In this iterative process, a discussion is divided in message subsets; each subset can be reconstructed separately, using a different model. Each intermediate reply-relation structure can be stored assuring that it stays unaffected in the consecutive execution of another model.
- **C3:** We use a *parent-child space* visualization to provide a visual evidence why particular message-pairs are classified as reply-relations, which considers the **R3** and supports the **RC3**. There, an overview of all computed reply-relation candidates using particular model encodes the reliability of the extracted structure (e.g., the more parent candidates are present, the less reliable is the extracted structure).
- **C4:** To satisfy the **RC4**, we use already known visualization methods, such as thread arcs [33], to represent the threaded structure. We expand these methods with multiple transition techniques to disentangle discussion's subtopics.

1.4 STRUCTURE OF THE THESIS

The thesis is built up the following way: Chapter 2 presents the related work to the thread structure's reconstruction methods and visualization techniques. Features, which are used to reconstruct the reply-relations, are presented in Chapter 3. The supervised classification models are explained and evaluated in Chapter 4; the unsupervised query based model is described and evaluated in Chapter 5. Chapter 6 presents the used visualizations to show the certainty of the extracted structure, and to provide more insights in discussion's content. The pipeline describing the models' execution, and the linkage between different visualizations is explained in Chapter 7. Chapter 8 shows the summary of the evaluation results, and presents multiple use cases. Chapter 9 shows the conclusion of our approach, and Chapter 10 lists possible improvements and future work.

2 RELATED WORK

In this chapter, we introduce previous research efforts related to the *RTS* task and frequently used visualization techniques representing threaded data. Thread structure has been advantageously applied to many different research problems [65], such as text classification task in discussion forums [66], or newsgroup search [69].

In general, the reconstruction of thread structure can be done based on two approaches: unsupervised and supervised techniques. In unsupervised techniques, the relation is found, by first, calculating the similarity between two messages which is considered as a relation weight. Then, other structural features can be used such as *distance* between messages to adjust the relation weight [4]. In supervised techniques, a learning algorithm is used to classify reply-relations. There, a model is trained using feature vectors. The trained model can be used to classify relations in the test data.

2.1 FREQUENTLY USED FEATURES

In the prior work, multiple features are combined to extract the reply-relation structure. Usually, features are divided into two main groups - semantic and structural features.

SEMANTIC (TEXTUAL, INTRINSIC) FEATURES Semantic features are used to extract message pairs with a similar content. The *cosine similarity* function is most commonly applied to calculate the similarity between two messages. This function measures the cosine of the angle between two input vectors. Frequently, input vectors are expressed as TF-IDF (Term Frequency - Inverse Document Frequency) [58] values. Multiple authors [3, 66, 64, 52] suggest to preprocess the data, before the function is applied. Stop words can be filtered out to reduce the dimensions of the vectors. Lemmatizer or stemmer can be used to generalize the tokens.

Yeh et al. [71] calculate a *unigram overlap* to measure the similarity between two messages. This feature is computed as the number of unique shared words between two messages, divided by the total number of the union of unique words in both two messages.

Author's language model can be used for discussions, where users actively participate in the conversation. In order to take advantage of this feature, messages of each discussion's participant are appended to each other. Then, similarity is calculated between the joined messages of two participants [4].

Reference to author's name may be a very reliable feature if present in the thread data. If the name of a discussion's participant appears in the body of a message, then all previously written messages by this participant can be seen as possible parent

candidates [4, 64, 51]. Liu et al. [38] use a person resolution for the patient forums; authors distinguish not only if one person is mentioned in the message, but try to find matches for person roles, second and third person pronouns.

Discussion participants may quote previous messages. *Quote* is one of the most frequently applied features to reconstruct the reply-relation structure [3, 13, 52, 71]. It is a highly reliable feature, but not present in all conversational text datasets.

STRUCTURAL (NON-TEXTUAL, EXTRINSIC) FEATURES Structural or non-textual features differ from the semantic features. These features don't take the content information into account, but observe an additional information like the position of the message in the discussion.

One of the most frequently used structural features is the *reply distance*. The *distance* between two posts expresses how many other messages have been posted in time between. Some authors calculate the likelihood that a post with location index i_1 is a parent post of a child post with location index i_2 [52]. The *length of a thread* can be used in a combination with the *distance* feature [3].

Time-distance between two messages (or *recency*) may also express the existence of a reply-relation [4, 64, 52]. This feature is useful if timestamps are explicitly given in the data.

Usually, a forum participant does not reply to his previously written messages. Therefore, a binary feature can be used to describe if a message is posted by the *same author* [64].

2.2 FREQUENTLY USED EVALUATION METRICS

A selection of a suitable evaluation metric is an important key for discriminating and obtaining the optimal classifier [27]. The most frequently used metrics in the related work are *accuracy*, *precision*, *recall* and *F-score*.

CONFUSION MATRIX A *confusion matrix* is often used to describe the performance of a classification model on a set of test data, for which the true values are known. Each column of the matrix represents the instances in a predicted class, while each row represents the instances in an actual class (or vice versa) [44].

	Predicted: true	Predicted: false
Actual: true	True Positive (TP)	True Negative (TN)
Actual: false	False Positive (FP)	False Negative (FN)

ACCURACY Accuracy is the most used evaluation metric in practice either for binary or multi-class classification problems. Through accuracy the quality of produced solution is evaluated based on percentage of correct predictions over total instances. The accuracy has several weaknesses which are less distinctiveness, less discriminability, less informativeness and bias to majority class data [27].

$$accuracy = \frac{TP+TN}{TP+TN+FP+FN}$$

PRECISION Precision is the fraction of relevant instances among the retrieved instances [44].

$$precision = \frac{TP}{TP+FP}$$

RECALL Recall is the fraction of relevant instances that have been retrieved over total relevant instances in the data [44].

$$recall = \frac{TP}{TP+FN}$$

F-SCORE F-score can be interpreted as a weighted average of the precision and recall [44].

$$f - score = 2 \cdot \frac{precision \cdot recall}{precision+recall}$$

COHEN'S KAPPA Although Cohen's Kappa measure is not used by authors of the related work, it can be an important indicator of the model's quality. Agreement statistics argue that accuracy does not take into account the fact that correct classification could be a result of coincidental concordance between the classifier's output and the label-generation process. The Cohen's Kappa measures inter-rater agreement for qualitative items [46]. If p_a is the proportion of observations in agreement and p_e is the proportion in agreement due to chance, then:

$$\text{Cohen's Kappa} = \frac{p_a - p_e}{1 - p_e}$$

2.3 UNSUPERVISED METHODS

Wang et al. [66] use a simple graph-based representation of a collection of messages where connections between messages are postulated based on inter-message similarity. This similarity is calculated using *cosine similarity* function of TF-IDF vectors. They use three different penalization functions to remove edges between nodes which do not satisfy the selected function (e.g., the *time-distance* between messages). The used dataset contains a collection of messages contributed within an educational teaching tool called LegSim. It has no given ground truth structure. Thus, the reply-relations are manually reconstructed based on their observed semantic relationship and explicit discourse markers. Leave-one-thread-out cross-validation [48] methodology is used to evaluate the model. The evaluated dataset has threads of length till 70 messages. For the longest threads, the F-score of the model is between 0.3-0.4. The model performs best on short threads (2-5 messages per thread); it reaches an F-score of 0.6. Obviously, the *cosine similarity* and penalty functions alone can not guarantee a high model's performance (especially on relatively long threads). Authors say that not only the thread

length, but also the average length of messages influences the model's performance. Overall, the performance tends to increase when the average message length increases.

Lin et al. [37] present a sparse coding-based model named SMSS (Simultaneously Model Semantics and Structure). The model projects each message into a topic space, and approximates it by a linear combination of previous messages in the same discussion. The precision, recall and F-score of the model are 0.52. Authors emphasize the need to identify *junk* messages. When *junk* messages are excluded from the dataset, the performance of the model increases more than 20%. Authors work with two different datasets - Apple and Slashdot - both having relatively long messages (in average more than 70 words per message).

Yeh et al. [71] present two approaches to link messages by their parent-child relations in an e-mail thread. The first approach uses e-mail header information, the second uses string similarity metrics, measuring the similarity between the quoted part of a child message and the unquoted part of a parent message. They use *unigram overlap* feature, which is computed as the number of unique shared words between two messages, divided by the total number of the union of unique words in both two messages. The average recall of their second approach is greater than 0.9. One needs to consider that the model performs well on relatively short threads (in average three e-mails per thread) where the *quote* is a very frequent feature.

Domeniconi et al. [14] try to solve a more general problem; authors analyze if two messages belong to the same thread (not explicitly linking them with a reply-relation). They use an unsupervised and a supervised method to distinguish threads from conversational texts. They map each message into a three dimensional representation based on its semantic content, the social interactions (in terms of sender and recipients) and its timestamp, and apply DBSCAN [11] clustering technique to extract threads. Additionally, they propose a supervised technique to classify message pairs belonging to one thread. Authors use a balanced dataset to create and test a Random Forest [6] model. This method does not reconstruct the reply-relations though, therefore the results are not comparable with our method.

2.4 SUPERVISED METHODS

Most of the work to reconstruct the thread structure has been done using supervised methods.

Schuth et al. [51] use multiple features to train a Decision Tree [53] classifier to reconstruct the reply-relations of news-comments. Used features are based on the *reference to author's name*. Although the F-score of their classifier by combining multiple features reaches almost 0.82, this method can be used only on datasets, where the author of the parent message is explicitly mentioned in the child message. Balali et al. [4] explore this method on different datasets (comments of news articles) and state that this model has good results only on news that have less than 20 comments because in these news articles most comments reply to the root.

Aumayr et al. [3] use 5 features (*cosine similarity*, *distance*, *time difference*, *quote*, *thread length*) and train SVM [10] (Support Vector Machine) and Decision Tree models. For training and testing the model, they use equally distributed set of "positive" and "negative" samples, with 213,800 pairs of posts. They limit the length of used threads; a thread should be at least three messages, and at most 40 messages long. Using three features (*cosine similarity*, *distance*, *quote*) their model reaches very high (0.92) F-score. Although it seems to be an outstanding result, the used dataset has almost 78% relations, where a reply follows immediately after the post it is responding to, and 20% of responses use *quotes*. It means that by default the lowest accuracy of the model is very high. Reply-relation *distance* of 1 is not present in many discussion datasets.

Seo et al. [52] use intrinsic and extrinsic features, and ranking SVM [25] to reconstruct the reply-relation structure. Intrinsic features describe the content similarity of two messages (including *quotes*), the extrinsic features contain information like *time-distance* between messages, or *reference to author's name*. They use three different forum datasets; they manually annotate 60 threads for World of Warcraft (WOW) and Cancun forums and use 1635 threads from W3C e-mail dataset (containing at least three e-mails). To test the model, they use a 10-fold cross-validation method on the first two datasets and take 100 threads for the testing for the third dataset. The recall for each of the datasets is 0.8798, 0.6279, 0.9617 respectively. The first two results are based on the manually annotated training corpus. It is hard to compare results of manually annotated corpus with corpus having the ground truth data though, as annotators rely on the apparent content similarity of messages. The third corpus has relatively short e-mail threads. Authors say that their approach shows a good performance for thread structure discovery when features which they introduce are available [52]. WOW, and W3C forums have better results due to the presence of *quotes*.

Balali et al. [4] use a similar method as [52], but they exclude the *quote* feature from their feature set. After extracting parent candidates, using SVM, they use heuristics like "a participant does not reply to the root post in his/her second comment" or "a participant does not reply to his own messages" to extract one most probable parent message. Authors test their model on two different datasets, the highest accuracy (which in this case is equal to precision, recall, and F-score) of the model is 0.7 for threads having less than 20 messages. For longer discussions (100-180 messages), the accuracy decreases to 0.3-0.4. Not only the performance of the model is not satisfying for longer discussions, but it is problematic to use this model on discussions where participants are not active. Features like *author's language model*, *author's activity*, which are part of the used six features, would not be useful for discussions with inactive members.

Wang et al. [64] propose a probabilistic model (threadCRF) to predict the replying structure for threaded discussions. They use two groups of features: node and edge features. Node features depend on the observed attributes in a post. Edge features are defined over two parent assignments together with the observed attributes in nodes. Authors use 31838 threads for the evaluation. However, only about 400 threads are longer than ten messages. That means, the model is applicable on short discussions, and might not perform the same on longer data. Authors define a new set of metrics which are used to evaluate the model.

Liu et al. [39] use the threadCRF to extract reply-relation structure from patient forums. Patient forums differ from other forum data, as the person reference relationships are critical to understand discussion's context. For example, when one post replies to another, the child post tends to include the person described in the parent post. Therefore, authors use such features like matching between the *address*, the *signature*, or the *role* of the person (e.g., my daughter), which is specific for the used dataset and can not be generalized for other forum discussions. To evaluate their model, authors use 200 threads, and accuracy as the only evaluation metric. The accuracy of their model is 0.635.

Dehghani et al. [13] reconstruct a linear and a tree structure of e-mail conversation threads. They use features like *content similarity*, *named-entity similarity*, or *speech act*. To rebuild the linear structure, they execute three steps. In the first step, a network indicating the relationships between e-mails is created. In the second step, this network is clustered (using Personalized Pagerank clustering method [61]) into conversation threads. In the third phase, e-mails in each group are arranged in chronological order to reveal the linear structure of conversation threads. To reconstruct the tree structure, they employ a learning-to-rank approach to learn a model to find an argument path from each node to the root of the conversation tree. Although the model reaches high recall of almost 0.9 for all used datasets, the average length of threads they work on are only 6-12 e-mails.

Huang et al. [28] extract <title message - reply> pairs to use this information for chatbox conversations. Replies which are logically relevant to the thread title are extracted with SVM classifier, based on message structural and content correlations. The use case differs from ours, as the authors aim to find most representative answers to the title message and do not reconstruct the whole threaded discussion.

2.5 SUMMARY OF OBSERVED MODELS

An overview of observed models which reconstruct the reply-relation structure is shown in Table 1, presenting the used algorithm, the best evaluation results and possible reasons, why the model could reach particular results. Half of all approaches analyze datasets having frequent reliable features (e.g., *quote*, *reference to author's name*). Most of the supervised approaches work on relatively short discussions (3-12 messages per thread in average). Two models are tested only on manually annotated data. All these reasons impact the evaluation results, and indicate, that the model's performance is highly dependent on the used dataset. Thus, most of the models have very restricted applicability.

2.5.1 *Imbalanced Classes Problem*

Previous section shows that most of the supervised models are trained and tested on a relatively short dataset. Some authors set a limitation, that the thread should be at least three messages long to be used for training the model [52]. None of the mentioned

Ref.	U/S*	Algorithm	Prec.	Rec.	F-sc.	Acc.	Characteristics
[66]	U	graph-based	-	-	0.7	-	**long messages** (avg > 60 words) **educational discussions**
[37]	U	SMSS	0.524	0.524	0.524	-	**long messages** (avg > 70 words) **manually annotated data**
[71]	U	similarity matching	-	0.8739	-	-	**reliable feature** (*quotes*) **e-mails** **short threads** (avg three e-mails)
[51]	S	Decision Tree	0.8307	0.6638	0.7379	-	**reliable feature** (only one feature: *reference to author's name*) **manually annotated data** **short threads** (4-comment threads)
[3]	**S**	**Decision Tree**	**0.939**	**0.918**	**0.928**	**-**	**reliable feature** (79.7% of the replies have a *distance* of 1) **balanced training dataset** **3-40 posts per thread**
[52]	S	Ranking SVM	-	0.9617	-	-	**reliable feature** (*quotes* as one of the main features) **e-mails** **short threads** (at least three e-mails)
[4]	S	SORTS: Ranking SVM + candidate filtering	0.5264	0.5264	0.5264	-	**long messages** (avg 63.4 words)
[13]	S	PPC + Ranking SVM	-	-	-	0.970	**e-mails** **short threads** (avg 6-12 e-mails)
[39]	S	threadCRF	-	-	-	0.635	**reliable feature** (*reference to author's name*, *person resolution*)
[64]	S	threadCRF	-	-	-	-	uses own set of metrics **short threads** (avg 6 messages)

Table 1: Summary of algorithms which are used to reconstruct the reply-relation structure. Listed are the **best evaluation results** of each paper, and the reasons, why these results could be achieved. The best performance is reached by [3], using Decision Tree algorithm. (* U-unsupervised, S-supervised, Prec.-precision, Rec.-recall, F-sc.-F-score, Acc.-accuracy)

authors work explicitly on relatively long threads (e.g., 100 messages per discussion and more). Some authors evaluate threads of different lengths and show that the performance of the classification model is always better for shorter threads. Wang et al. [64] emphasize that the size of a thread influences the model's performance; if the thread length increases, the performance of the model is reduced.

Often real-world datasets are predominately composed of "normal" examples with only a small percentage of "abnormal" or "interesting" examples [8]. That is known as the *imbalanced classes problem*. Classification of message reply-relations belonging to one relatively long thread is a representative example of this issue. If a thread discussion contains n messages, then at most $n-1$ reply-relations may exist in data, under the assumption that one message can have at most one single parent. At the same time, $\frac{n(n-1)}{2}$ false reply-relation candidates exist. If n is relatively large, then the two reply-relation classes - "positive", and "negative" - are highly imbalanced (e.g. for 100 messages, $\approx$ 2% reply-relations are of the class "positive" and $\approx$ 98% - of the class "negative").

Usually, the classification algorithms (for two-class or multi-class problem) require the data to be balanced, meaning, that there should be the same (or similar) amount of instances representing different classes. This requirement influences the classifier's performance significantly. The performance of machine learning algorithms is typically evaluated using predictive accuracy. However, this is not appropriate when the data is imbalanced, as the accuracy of the model could reflect the underlying class distribution [68]. That means, the model is very likely predicting the majority class regardless of the data it is asked to predict.

Various methods exist to deal with the imbalanced classes problem. Firstly, one can resample data using an under-, or over-sampling technique to create an artificially balanced training dataset. From all the papers discussing the thread structure's reconstruction task, only [3] and [14] describe, how the training and test datasets are distributed. Authors use an equal number of "positive" and "negative" instances. Secondly, a weighted classification algorithm can be applied, where the weights for each of the classes can be customized, so the best split between two classes is created [29]. Thirdly, one can use an ensemble technique [57]. Ensemble methods combine multiple learning algorithms to obtain better predictive performance than it could be achieved from any of the constituent learning algorithms alone. Fourthly, depending on the classifier's task, one can specify costs for false positive classified instances for different classes; the precision or recall for one particular class may be improved.

UNDER-, OVER-SAMPLING Most of the classification algorithms demand a balanced training dataset to learn a model. If the original training data is imbalanced, one can use under-, or over-sampling technique to artificially balance it. In the under-sampling technique, instances of the majority class are reduced to the number of instances of the minority class. An important drawback is that this reduction might discard potentially useful data. In the over-sampling, instances of the minority class are increased to the number of majority class, by using exact copies of existing instances. Such replication makes the overfitting of training data likely [68]. Japkowicz

[31] shows that over-sampling with replacement doesn't significantly improve minority class recognition.

SYNTHETIC MINORITY OVERSAMPLING (SMOTE) Chawla et al. [8] present that a combination of over-sampling the minority class and under-sampling the majority class can achieve better classifier performance than only under-sampling the majority class. They suggest a method called SMOTE to oversample the data. This method involves creating synthetic minority class examples. Authors say that it outperforms traditional over-sampling because in the over-sampling the instances are just replicated. That, however, means that the decision region that results in a classification decision for the minority class can become smaller and more specific. That is the opposite of the desired effect. In contrast, the synthetic over-sampling causes the classifier to build larger decision regions that contain nearby minority class points [8].

WEIGHTED CLASSIFIER For weighted classifiers, the penalty of misclassification for each training sample is distinctive. By setting the equal penalty for the training samples belonging to the same class, and setting the ratio of penalties for different classes to the inverse ratio of the training class sizes, the obtained model can compensate for the undesirable effects caused by the uneven training class size. Huang et al. [29] use Weighted SVM to train a model. Authors state that not only the performance of the model on different classes is better than when training the model with unequally distributed classes, but also the classification accuracy for the class with a small training size is improved.

ENSEMBLE ALGORITHMS Ensemble learning trains multiple classifiers and selects some of them for an ensemble. The combination of multiple classifiers can be more effective compared to the individual ones [57]. Ensemble learning is implemented as two levels. The first level consists of training weak base classifiers, the second level - selectively combining the member classifiers into a stronger classifier. Members of an ensemble may be constructed, by applying a single learning algorithm, or by using different learning algorithms over a dataset [17].

COST SENSITIVE CLASSIFIER Cost sensitive classification technique incorporates the knowledge that the expense of false positive instances may vary for different classes. The user sets costs for false positive instances of different classes; the classifier tries to minimize the overall cost. This method introduces biases so that certain kinds of errors are avoided when those errors incur in a high cost [18, 15].

2.5.2 *Why Visual Analytics?*

None of the related work talks about the imbalanced classes problem. It is possible, that authors don't need to deal with this obstacle, if relatively short threads are used to train and test the model. For example, if a thread has 10 messages, then the difference between "positive" and "negative" classes is much smaller ("positive": $\approx$ 15% and "negative": $\approx$ 85%) than if a thread has 100 messages ("positive": $\approx$ 2% and "negative": $\approx$ 98%). However, models [3, 14], which use a balanced training data, might meet chal-

lenges, applied on a new unseen test data with a highly imbalanced distribution of "positive" and "negative" reply-relation instances. The performance might decrease, as the models could have learned erroneous information, not existing in the real world's data.

The quality of the used dataset also influences model's performance significantly. If no reliable features (e.g., *quotes*) are present in the used dataset, which are descriptive for the reply-relations, then there is a higher chance that the model will overfit the training data. Besides, most of the related work deals with the thread structure's reconstruction task on only one specific discussion's type using only features present in the data to create a model. That means, the user has to apply multiple models on different conversational text datasets.

Challenges like imbalanced classes problem or model's overfitting due to a bad quality of data show that sometimes the performance metrics like precision and recall may be misleading. In such situations, a visual analytics workspace is highly beneficial. It may provide insights, which relations may be reliably reconstructed and which features influence the reconstruction's process most. The workspace may also be helpful to deal with the specificity of existing models, as it is possible to improve the model's performance by incorporating user's knowledge.

Being able to explore the decisions a model makes and identifying potential issues is crucial in application areas where experts need to build trust in model's decisions [34]. A common practice is simply to focus on model's accuracy (or other evaluation metrics presented in previous sections), to get a sense how good the model performs. Kraus et al. [34] emphasize that such statistics do not provide insights on how or why a model fails to capture important phenomena accurately. Thus, many researchers have voiced the need for more transparency in model's made decisions when the application domain requires it [42, 40, 21, 41, 7].

So far we know, a visual analytics tool which provides insights into the reply-relation reconstruction process, does not exist.

2.6 VISUALIZATION OF THREAD STRUCTURE

Some prior work describe capabilities to visualize the given thread structure of a conversational data. Authors aim to provide a better overview of the known reply-relation structure.

Frequently, multiple *thread arcs* [33] techniques are applied to visualize a thread structure. Thread arcs method represents discussion as a tree, where messages are shown as nodes and reply relations are displayed as edges between nodes. There, the chronology of messages is combined with the branching tree structure of a conversational thread.

Fu et al. [23] extend basic thread arcs technique and present *thread river*, which can illustrate temporal and structural information of lengthy threaded discussions. In their tool called iForum, they offer a set of visualization designs for presenting the

main interleaving aspects of MOOC forums at three different scales (posts, users, and threads).

ForumReader [12] is a tool combining visualization techniques with automatic topic extraction algorithms to help users explore Flash forums. ForAVis [67] integrates sentiment analysis. Hoque et al. [26] have developed ConVis to support multi-faceted exploration of blog conversations, which contains multiple views to provide thread information at different granularities.

Some authors use visualization techniques to show topic changes within a discussion, like Trampus et al. [62], or Liu et al. [38] by applying the river metaphor to show content changes over time. In [62] a semantic "atlas" provides a thematic overview of larger forum segments, and a timeline displays the temporal evolution of forum topics. Liu et al. [38] connect the corresponding topics at different times, and provide an overview of the evolving hierarchical topics. Additionally, a sedimentation-based visualization enables the interactive analysis of streaming text data from global patterns.

Like previously mentioned, no visual analytics tool is found, where authors, first, extract the reply-relation structure and then visualize it, providing feedback, which relations could be found reliably and why. The linkage between these two tasks - thread structure's reconstruction and visualization - is necessary though. Visual feedback on extracted structure's certainty may increase user's trust in the system.

3

DESCRIPTIVE FEATURES FOR THE REPLY-RELATION RECONSTRUCTION

This chapter presents the results of the initial data analysis, and describes the validity and reliability of 17 observed features to reconstruct the reply-relation structure of threaded data. Although some differences in the reliability of observed features are detected, they all remain in the final feature set. The extraction of features is done, using Java programming language.

3.1 INITIAL DATA ANALYSIS

10 random Reddit-threads are selected having a length of 100-200 messages to explore their message characteristics. Messages are categorized depending on their properties. The average length of a message in a thread is 34 words. Messages having less than 10 tokens are classified as short, messages having more than 40 tokens - as long.

Not all messages have a valuable content, such messages are called *junk*. Lin et al. [37] write, that a discussion thread usually focuses on a limited number of topics, while *junk* posts usually have different topics and act as outliers. Authors state that it might be necessary to extract these messages and remove them from the dataset to improve the reply-relation reconstruction's quality.

Multiple characteristics are used to distinguish if one message is a *junk*. Firstly, messages with less than three tokens are classified as *junk*. Secondly, in some cases, participants of the discussion may write messages which are not related to the discussed topic. Multiple topic modelings are used to distinguish such messages: LDA [5] (Latent Dirichlet Allocation), IHTM [2] (Incremental Hierarchical Topic Modeling), SWB [9] (Special Word Background Topic Modeling), and BTM [70] (Biterm Topic Modeling). If none of the used topic modelings can classify a message belonging to one topic with a high probability, then this message is classified as *junk*. Thirdly, sometimes, people in a discussion get offensive. Messages, where most of the words are banned words, are also classified as *junk*. In average, 9% of all messages in the explored threads are *junk* messages.

As mentioned in Chapter 2, different features may be used to extract the thread structure. One of the features is the *author's lexicon*. Unfortunately, 50% of participants in Reddit write only one message within one thread. Only 17% write more than three messages, therefore it is hard to use *author's lexicon* as a reliable feature for the Reddit data.

Frequently, messages in a forum discussion contain noise. If the main part of tokens in the message consists of special characters, then the content information in this message is limited. Thus, it might be difficult to classify the parent-child relations

correctly. 1% of all messages have this characteristic. If a message mainly contains only URLs, it might also influence the classification of its parent-child relations negatively. 2% of all messages contain only URLs.

12% of all messages have quotation marks. Frequently quotation marks are used to cite a previous message. Quotes may be used to represent irony, or metaphor too. 8% of all messages end with a question mark, which suggests searching for an answer in the following messages.

3.2 FEATURE CHARACTERIZATION

Both, unsupervised and supervised methods, use a set of features to reconstruct the thread structure. In the related work, features are divided into semantic and structural ones. We split all features into three main groups: content, structural, and meta-data features. Content features represent the content similarity of two messages. Structural features show, if two messages have the same structural elements, like the same *named-entities*, *quotes*, *substrings*. And meta-data features describe an additional information like the position of a message in the discussion.

3.2.1 *Content Features*

COSINE SIMILARITY  Two messages representing a reply-relation may have similar content. To extract such relations, one might use content features, such as *cosine similarity* [55]. The *cosine similarity* uses a bag-of-words (BoW) representation. BoW describes a textual message m by means of a vector $W(m) = w_1, w_2, ...$, where each entry indicates the presence or absence of a word w_i. The *cosine similarity* between two vectors m_i, m_j is measured by:

$$\cos(m_i, m_j) = \frac{W(m_i) \cdot W(m_j)}{||W(m_i)|| \cdot ||W(m_j)||}$$

Usually, words in these vectors are weighed by TF-IDF values. If tf describes the term frequency in a message, df describes the number of messages containing this particular term, and N represents the thread length, then TF-IDF [49] is calculated by:

$$\text{TF} - \text{IDF} = \frac{tf}{tf + 0.5 + 1.5 \cdot \frac{\text{message length}}{\text{avg message length}}} \cdot \log(\frac{N}{df})$$

Before applying TF-IDF function, we reduce the dimensionality of the vectors by filtering out stop-words. Due to the high level of noise in the forum data, tokens having only one character, are removed from the corpus vector. Lemmatizer is used to generalize the tokens.

As mentioned in the previous section, many messages in a thread are relatively short. Thus, we use multiple message enrichment techniques to enhance their content.

WORDNET ENRICHMENT Messages relating to the same subject may not include identical terms; they may, in fact, include words that are in the same semantic category. Thus, a dictionary (Wordnet[1]) is used to enhance tokens with their synonyms. Messages are enriched only with synonyms which are already present in the corpus vector, to avoid an overload of irrelevant terms. These enhancements are weighted lower than the original tokens.

URL ENRICHMENT A web-crawler is used to extract article content, being linked in the message. Only those tokens which are already present in the corpus vector are added to the message's word-vector. These additional tokens are weighted lower than the original message tokens.

TOPIC ENRICHMENT A topic modeling is used to enrich the message with its most significant topic keywords. Altogether four topic modelings are used, from which the user may select one for this task. These topic modelings are LDA, IHTM, SWB, and BTM. LDA learns latent topics in a corpus by exploiting document-level word co-occurrences. Hence, it typically suffers from data sparsity (estimating reliable word co-occurrence statistics) when applied to short documents [35]. Also, IHTM topic modeling is more suitable for long text data. More appropriate topic modelings for short messages are SWB and BTM. SWB is based on LDA. However, it allows words in documents to be modeled either from general topics, or from post-specific "special" word distributions, or from a thread-wide background distribution. This topic modeling is used in [37] as one of the baseline models to reconstruct the thread structure. BTM is the standard topic modeling for short text data, and from all four models it performs best, by correctly recognizing which messages belong to the same topic. We weight the topic-keywords lower than the original message tokens. Working with topic modeling is not trivial though, as no information about the number of subtopics in the data is given. The user has to guess the appropriate number of topics to be extracted.

AUTHOR ENRICHMENT Additionally to the previously mentioned four topic modelings, an ATM [50] (Author Topic Modeling) is used to enrich messages with keywords, representing the lexicon of the message's author. ATM is based on LDA, but it extracts the author-topic distribution instead of the document-topic distribution. As it is mentioned in the previous section, in average 50% of all authors in Reddit write only one message in a thread. Therefore, frequently author-keywords may not describe the author's lexicon correctly if not enough written text by one author is present. But it might be a reliable feature for other datasets, having more active discussion participants.

WORD EMBEDDING Coreference resolution is used to create a *word embedding* feature. Coreference means that two or more expressions in a text refer to the same person or thing [36]. We use it to find messages, which might be linked having the same referent. In the *word embedding* feature all mentions from one coreference-chain

1 https://wordnet.princeton.edu/

are replaced by the first mention (referent) of it. This feature has some flaws. First of all, the coreference resolution assumes that messages in a discussion are in a correct logical order. The exact position of a mention is used as an indicator for a possible coreference. Another problem is the quality of the coreference resolution itself used on a short and noisy forum data. Too many false positive mentions are found.

After each enrichment technique, the *cosine similarity* of the enriched message pairs is calculated anew.

TOPIC AGREEMENT Previously mentioned four topic modelings (LDA, IHTM, SWB, BTM) are used to explore their *topic agreement*. For that, the document-topic distributions of previously mentioned four topic modelings are observed. If at least k (a threshold set by the user) topic modelings agree that two messages belong to the same topic, then this reply-relation has a *topic agreement* feature. The quality of this feature, like the quality of other features using topic modelings, depends on the user's selected parameters (e.g., the number of extracted topics).

3.2.2 *Structural Features*

Seven structural elements (*quotes*, *substrings*, *n-grams*, *named-entities*, *lexical episodes* [24], *coreferences*, and *references to author's name*) are observed, to evaluate their descriptiveness for the *RTS* task. In some situations, these elements may overlap. We split these features apart to make better decisions regarding which information and which features are more reliable to reconstruct the thread structure. Each of the structural features describe a number of distinct elements (e.g., number of distinct *named-entities*) which are present in the parent and child message.

QUOTE Many online community systems support an option to quote a text from the preceding message when a message is uploaded. *Quotes* can be seen as a very accurate reply indicator. They are extracted, using regular expressions, where the quotation marks or ">" (greater than) characters are detected, and the body of the *quote* is compared to the content of all previously written messages to detect a match.

SUBSTRING In situations, when no explicit quotation marks are used to cite one of the earlier written messages in a discussion, a long common *substring* (having at least four tokens) of two messages may indicate a contextual connection. Common *substrings* are extracted for each message pair. At least four tokens long *substring* is a valid feature for the particular reply-relation candidate. The currently used minimum length of the substring is a heuristic and could be treated as a parameter.

N-GRAM If an *n-gram* is frequently mentioned in the discussion, it could be an indicator that these tokens are important for the conversation's primary subject. Thus, we extract frequent *n-grams* and observe their presence in each message pair.

NAMED-ENTITY *Named-entities* are concepts used in the NLP to refer to words for which one or many rigid designators stand for the referent. They can be used to group texts discussing the same topic [22]. The Stanford Named-Entity Recognizer[2] is used to extract them. The number of distinct *named-entities* is counted for each message pair.

LEXICAL EPISODE "*Lexical episodes* are portions within the word sequence of texts where a particular keyword appears more densely than expected from its frequency in the whole word sequence." [24] If the same word or *n-gram* appears densely in the discussion, it might indicate, that messages, where the keyword appears, are part of one subtopic. A position of the keyword in the discussion is used to distinguish if it is part of a *lexical episode*. The quality of extracted *lexical episodes* is arguable, as the temporal message order differs from the logical (reply-relation) order. Thus, the algorithm might detect *lexical episodes*, which do not exist in the original reply-relation structure.

COREFERENCE *Coreferences* are used not only to create a *word embedding* feature but also as a separate structural feature. Messages from the same mention-chain might indicate that they are part of one subtopic. Stanford CoreNLP[3] is used to extract *coreferences*. Similarly to *lexical episodes*, the quality of *coreferences* is arguable, due to the differences in the discussion's temporal and logical order.

REFERENCE TO AUTHOR'S NAME The *reference to author's name* feature means that the name of the parent message's author is explicitly mentioned in the child message. Like it is shown in Chapter 2, many existing works use the *reference to author's name* as an important feature to extract reply-relations. Similarly to *quotes*, this feature can be seen as an accurate reply indicator.

3.2.3 *Meta Data Features*

Three meta data features are observed regarding their suitability to reconstruct reply-relations. Those are: *distance* between messages, *time-distance* between messages, or two messages having *different authors*.

DISTANCE The *distance* between two posts expresses how many messages have been posted in the time between them. The values of this feature depend though on the data source. In different forums, the participants may observe the messages in a chronological or non-chronological order. That may influence the behavior of the forum users. Thus, for different forums, the average *distance* between messages may, in fact, vary.

2 https://nlp.stanford.edu/software/CRF-NER.shtml
3 https://stanfordnlp.github.io/CoreNLP/coref.html

TIME DISTANCE In the related work, multiple authors use the *time-distance* feature. They say that usually there is a short time-span between replying messages in a thread. Some topics are more actively discussed than others. Thus, depending on the popularity of the topic, the *time-distance* values for different discussions may vary.

DIFFERENT AUTHORS Participants of a discussion do not usually reply to their previously written messages. If two messages have *different authors*, then it is a valid reply-relation candidate.

3.3 FEATURE ANALYSIS

10 Reddit files, mentioned in the previous section, are used to explore their feature distribution and feature descriptiveness in the real world's data. For content features, the minimum similarity level of two messages to be seen as a reply-relation candidate is 0.2. The minimum *topic-agreement* is 2 (e.g., at least two topic modelings have to assign the particular two messages to the same topic). Regarding the meta-data features, the maximum *distance* between two messages should be 1, and the maximum *time-distance*: 24 hours.

Table 2 shows the average frequency of individual features in the given reply-relations, and the average probability to extract them, taking into account their recurrence in all possible reply-relation candidates.

FEATURE FREQUENCY (VALIDITY) A feature is valid to be used for the reply-relation reconstruction task if it is present in the data. Regarding the frequency of single features in the given reply-relations, the *topic-agreement* and the *coreference* are two most present features in the data, excluding the meta-data features. Meta-data features for the used input data files are less descriptive, as none of the authors have replied to their previous messages, and all messages have been written in a time-span of 24 hours, after the creation of the thread.

As it is shown in Chapter 2, *quotes* and *reference to author's name* are used by multiple authors [4, 64, 51, 52] as the most relevant features to reconstruct the thread structure. Unfortunately, both of them are not very common in the Reddit data.

FEATURE PROBABILITY (RELIABILITY) A feature is reliable if it is descriptive for the given reply-relations. More reliable are those features, which are present in the given reply-relations, but infrequent in all other reply-relation candidates.

Features representing the content similarity are relatively reliable (e.g., *cosine similarity*). The best possible way to reconstruct the correct reply-relation is using *quotes* or *substrings*. The performance of *substrings* is slightly better than of *quotes*. Authors may use quotation marks not only to quote a parent message, but also to highlight, for example, a figurative meaning (a metaphorical, idiomatic, or ironic sense of a word in contrast to the literal meaning).

Feature	Frequency	Probability to Extract
Cosine Similarity	13%	**28**%
Url Enrichment	12%	**27**%
Topic Enrichment	10%	12%
Author Enrichment	9%	14%
Wordnet Enrichment	14%	**29**%
Word Embedding	12%	**28**%
Topic Agreement	**42**%	2%
Quote	7%	**50**%
Substring	6%	**55**%
N-Gram	10%	7%
Named-Entity	10%	5%
Lexical Episode	6%	10%
Coreference	**26**%	4%
Reference to Author's Name	0%	0%
Time-Distance	100%	2%
Distance	5%	5%
Different Authors	**100**%	1%

Table 2: Feature distribution in 10 Reddit discussion files. The frequency indicates how valid the feature is to be used for the reply-relation reconstruction. The probability shows, what is the chance to extract the reply-relation correctly, using the particular feature.

As it is mentioned before, one of the most frequent features in the given relations is the *topic agreement*. Unfortunately, the probability to extract relations correctly, using this feature, is very low (only 2%). That shows, the feature is present not only in the given reply-relations but also in false reply-relation candidates. That could be related to the selected number of topics for topic modeling algorithms. In this case, the parameter might be set too low, as too many message-pairs are found to be within the same topic.

4 MACHINE LEARNING

This chapter describes how machine learning algorithms can be used to reconstruct the reply-relation structure of a conversation data. As it is shown in Chapter 2, most of the related work deals with the thread structure's reconstruction task using supervised machine learning techniques. Different algorithms have been applied for this task, such as Decision Tree, Random Forest, SVM, Ranking SVM, and CRF. Authors present the performance of their created models, using different metrics like accuracy, precision, recall or F-score. The presented models perform well; the highest score is reached by Aumayr et al. [3] having precision, recall and F-score greater than 0.9.

4.1 CHOICE OF CLASSIFICATION ALGORITHM

It is difficult to distinguish the best algorithm used in the related work, as their performance depend on the used dataset and the presence of reliable features (e.g., *quotes*, *reference to author's name*).

Algorithms used in the prior work may be divided into two general groups: those, which see the reply-relation reconstruction as a sequence labeling task (e.g., CRF), and those, which treat each observation (message) independently (e.g., Decision Tree, SVM). Best performing models presented in Chapter 2 are part of the latter group. In order to select one algorithm from this group, we introduce a list of criteria. These criteria are:

1. No crucial hyper-parameter optimization should be needed.
2. It should be time efficient to train and test the model.
3. The overfitting should be avoided.

Regarding the first criteria, some algorithms like SVM can require multiple parameters, which govern the training process; these settings can have a profound affect on the resulting engine's performance [59]. On the contrary, Decision Tree algorithm is parameter-free [53]. Regarding the second criteria, Aumayr et al. [3] show that SVM is more time-consuming to be trained than, for example, Decision Tree. Also, the training of Random Forest takes longer than the training of Decision Tree model [47]. Regarding the third criteria, Random Forest has an advantage over Decision Tree, as it can limit overfitting without substantially increasing error due to bias [47].

After evaluating the criteria, Decision Tree and Random Forest algorithms are selected as appropriate ones for the thread structure's reconstruction task. Decision Tree is used by Aumayr et al. [3], having the highest performance in observed related work. For comparison, we train a second model using Random Forest algorithm, as it is more

robust against overfitting. The WEKA[1] framework is used to train and test the models. Random Forest model is created using default parameters (10 trees).

4.2 GENERATION OF TRAINING DATASET

In order to train and evaluate a supervised classification model, an appropriate training dataset has to be created. We use a data corpus provided by one member of Reddit community. This corpus contains all messages (comments) of Reddit discussions, written in July 2017, a total of 81,798,725 publicly available comments (7.78 GB compressed, 54.8 GB uncompressed). We extract first 1,000,000 comments in the temporal order to generate our training dataset. These messages are, first, joined in separate files, each representing one single thread. Threads which have less than three comments and more than 500 comments are removed. Besides, we remove such threads where no reply-relations are present.

The final dataset contains 6926 threads. They are used to create the training data instances. The existing reply-relations are extracted, and particular instances are labeled as "positive". All remaining relations (artificially created), which are not given in the thread, are labeled as "negative". One instance contains 18 features and the class variable ("positive" or "negative"). The numerical features (e.g. *cosine similarity*, *distance*) are simply added to the feature set. The boolean feature (e.g. *different author*) is normalized to integers. "1" is used, if the particular feature is present in the relation, and "0" - if it is not present.

The training dataset contains 116,154 "positive" instances, and 19,737,566 "negative" ones. Apparently, these two classes are highly imbalanced. From all instances, less than 1% are classified as "positive".

4.3 FEATURE SELECTION

The training of a good and reliable classification model requires a reliable feature set. Thus, to assure the maximum quality of the model, we do a feature selection to dismiss less-reliable features, which might not represent the conversation data properly.

UNRELIABLE FEATURES As it is shown in Chapter 3, the quality of some presented features for Reddit data is arguable. The *author enrichment* feature may not represent author's lexicon properly, as many discussion participants are inactive and write only one message in a thread. The quality of the *topic enrichment*, and *topic agreement* features depend on the number of topics selected by the user. Thus, it is difficult to automatically develop a reliable training data, using these features.

The *coreference* and *lexical episode* features assume, that the input data is in a logical order, which is not true. Thus, the quality of these features might influence the classification negatively. Hence, they are removed from the feature vector. The *coreference* feature is used to generate the *word embedding* feature. Thus, it is also removed from the final feature set.

1 http://www.cs.waikato.ac.nz/ml/weka/

ADDITIONAL FEATURES We use *thread length* as additional feature, which might help the classifier to make better decisions. *Thread length* shows how many messages are written in the thread. The distance between the current message and the previous message of the particular author is also used as an additional feature for the machine learning. It is called the *same author's distance* feature.

FINAL FEATURE SET The final feature set contains 13 features and a class variable. These feature are: *cosine similarity*, *wordnet enrichment*, *url enrichment*, *quote*, *substring*, *n-gram*, *named-entity*, *reference to author's name*, *distance*, *time distance*, *different author*, *same author's distance*, *thread length*.

4.4 PERFORMANCE OF THE MODEL TRAINED ON IMBALANCED DATASET

Due to the computation costs, we use a subsample of the original dataset which represents the distribution of the two classes ($\approx$ 2% of class "positive", and $\approx$ 98% of class "negative"). The training dataset has 10,000 "positive" instances, and 500,000 "negative" ones.

10-fold cross validation method is used to evaluate the model's performance. The results of the Random Forest model is shown in Table 3. Apparently, the model is negatively influenced by the majority ("negative") class. Although the accuracy of the model is 0.98, the precision of the "positive" (minority) class is 0.003, and the recall is 0.002. Hence, the model is useless for the classification of reply-relations. The Cohen's Kappa statistics shows it significantly - the result is -0.006, meaning that the probability of observed disagreement exceeds chance-expected disagreement.

Model	Precision (p)	Recall (p)	F-score (p)	Cohen's Kappa	Accuracy
Random Forest	0.003	0.002	0.002	-0.006	0.98

Table 3: Evaluation results of the Random Forest model, trained on an **imbalanced dataset**, tested using 10-fold cross validation technique. (*p - "positive" class)

4.5 DEALING WITH IMBALANCED CLASSES PROBLEM (UNDER-SAMPLING)

As it is described in Chapter 2, different techniques exist to deal with the imbalanced classes problem. Some of these techniques aim to generate an artificially balanced training dataset, reducing so the weight of the majority class.

We take the idea of the best performing model created by Aumayr et al. [3]. That is one of two works, where authors explicitly describe, how the training dataset is created. Authors don't talk about the imbalanced classes problem itself, but they do balance the amount of "positive" and "negative" instances in their training data, by randomly selecting the "negative" ones. Their dataset has threads of length from 3 to 40 posts, altogether 213,800 data instances. The model is trained using Decision Tree algorithm and evaluated using 5-fold cross-validation technique.

Similarly to [3], we artificially create a balanced training dataset. The under-sampling technique is used, to reduce the number of instances representing the majority class. The final dataset has 110,038 "positive" instances, and 110,038 "negative" ones (in total 220,076 instances).

In order to see how good the model of Aumayr et.al. [3] is applicable on other datasets having different data characteristics, we first train the Decision Tree and Random Forest models on 5 features (*distance*, *time distance*, *quote*, *cosine similarity*, *thread length*), which are used by [3]. Then, the rest of the features presented in Section 4.3 are added to the training dataset and models are trained anew. In such a way, the impact of the eight additional features in the learning process may be obtained.

The evaluation results of two models trained on different feature sets are shown in Table 4. Models using five features applied on Reddit data perform worse than the model of [3]. However, the performance is still solid. By adding the remaining eight features to the feature set, the performance of the models can be slightly increased.

Model	Precision	Recall	F-score	Cohen's Kappa	Accuracy
Decision Tree (5) by [3]	**0.939**	**0.918**	**0.928**	-	-
Decision Tree (5): Reddit	0.774	0.687	0.728	0.4733	73.6021
Random Forest (5): Reddit	0.714	0.671	0.692	0.402	70.0981
Decision Tree (13): Reddit	**0.793**	**0.684**	**0.734**	0.5053	75.2663
Random Forest (13): Reddit	0.740	0.695	0.717	0.4511	72.5572

Table 4: The first row shows the original evaluation **results of [3]** using Decision Tree algorithm. The following rows display the evaluation results of two models trained on a **balanced Reddit dataset using 5 features** (presented by [3]). And the last rows show the evaluation results of two models trained on a **balanced Reddit dataset using 13 features**, presented in Section 4.3.

4.5.1 *Common Feature Influence on Model's Performance*

Table 4 shows, that [3] reaches 20% higher F-score than our Decision Tree model, trained on five features. Thus, we compare results of *cosine similarity*, *quote*, *time distance*, and *distance* features separately, reached by Aumayr et al. [3] (shown in Table 5), and by Decision Tree model created on Reddit data (shown in Table 6). These results show how the presence of single features influence the final model's performance.

Feature	Precision	Recall	F-score
Distance	**0.938**	0.773	0.848
Time Distance	0.671	0.574	0.619
Quotes	0.981	**0.235**	0.379
Cosine Similarity	0.763	0.377	0.505

Table 5: Evaluation results of single features, presented by Aumayr et al. [3].

Feature	Precision	Recall	F-score
Distance	0.677	0.557	0.611
Time Distance	0.666	0.544	0.599
Quotes	0.946	0.037	0.071
Cosine Similarity	0.793	0.443	0.568

Table 6: Evaluation results of single features, using Decision Tree algorithm applied on Reddit data.

Tables 5 and 6 show that results of *time distance* and *cosine similarity* are similar for both models. Apparently, the dataset used by [3] have more *quotes*; the model reaches 20% higher recall using this feature. However, the largest difference in the precision is observed, when *distance* feature is used to train the model. Authors describe that their used dataset in 79.7% cases has a reply-relation *distance* of 1. Apparently, the performance of a trained classifier is highly dependent on the presence of reliable features in the used dataset.

4.5.2 *Performance on Unseen Imbalanced Test Data*

We use a representative sample of 40 Reddit discussions (described in Chapter 8), which are not included in the training dataset, to evaluate previously described Decision Tree and Random Forest models on an unseen imbalanced test data. The evaluation results are shown in Table 7.

Model	Precision	Recall	F-score
Decision Tree (5)	0.06	0.55	0.11
Random Forest (5)	0.05	0.56	0.08
Decision Tree (13)	0.07	0.54	0.12
Random Forest (13)	0.05	0.61	0.09

Table 7: Evaluation results of Decision Tree and Random Forest models, trained on a **balanced dataset**, tested on an unseen 40 Reddit discussions.

Although the observed models have a relatively good performance, tested using 10-fold cross-validation technique, they perform much worse on a new unseen data (the average precision reaches only 0.06). Such decrease in the performance can be caused by multiple reasons.

Firstly, the under-sampling technique has some drawbacks. When the data is balanced using under-sampling method, then many instances representing the majority class are removed from the dataset. Thus, some relevant information can get disregarded. Secondly, due to the absence of reliable features, the model is likely to overfit. Fig. 3 shows an example of the Decision Tree model trained on the *cosine similarity* feature, which overfits the training data. Apparently, no clear split exists between

"positive" and "negative" class instances for this feature. Thirdly, the size of the training dataset can also cause the model to overfit. Although altogether more than 200,000 instances are used to train the model, the data might be too small to describe a good split.

```
J48 pruned tree
------------------

COSINE_SIMILARITY <= 0.063368
|   COSINE_SIMILARITY <= 0.026734: false (154750.0/59398.0)
|   COSINE_SIMILARITY > 0.026734
|   |   COSINE_SIMILARITY <= 0.034615: false (3636.0/1765.0)
|   |   COSINE_SIMILARITY > 0.034615: true (11901.0/5037.0)
COSINE_SIMILARITY > 0.063368
|   COSINE_SIMILARITY <= 0.996492: true (49361.0/7547.0)
|   COSINE_SIMILARITY > 0.996492: false (428.0/197.0)
```

Figure 3: An example of the Decision Tree model trained on the *cosine similarity* feature, which overfits the training data.

Not only the artificially balanced data, or the absence of reliable features may influence model's performance, but also the length of single messages or the length of threads may have an impact on the reply-relation reconstruction's process. Thus, we observe the performance of the most frequently used features on different datasets, having varying thread length or varying average message length.

THREAD SIZE'S INFLUENCE ON MODEL'S PERFORMANCE To evaluate how the single feature performance is influenced by the thread size, we split all threads into six bins of different length: threads having 3-50, 51-100, 101-200, 201-30, 301-400, and 401-500 messages. An overview of the bin sizes is shown in Table 8. For each bin, a Decision Tree model is trained, using a balanced training dataset and one of the most frequently used features (e.g., *cosine similarity*, *quote*, *time distance*, or *distance*).

	3-50	51-100	101-200	201-300	301-400	401-500
threads	5000	1246	447	143	56	34
"positive"	16,273	38,897	29,166	15,453	8,993	7,372
"negative"	361,794	2,588,460	3,732,105	3,576,025	2,883,456	2,863,621

Table 8: An overview of datasets having threads of different length.

Using *cosine similarity*, *quote*, and *time distance* features, no significant changes on model's performance can be observed, when trained on different sized threads. The observations show that from all previously listed features, only the *distance* feature is influenced by the thread's size. The *distance* is more reliable for shorter discussions (3-50 messages), like Fig. 4 presents.

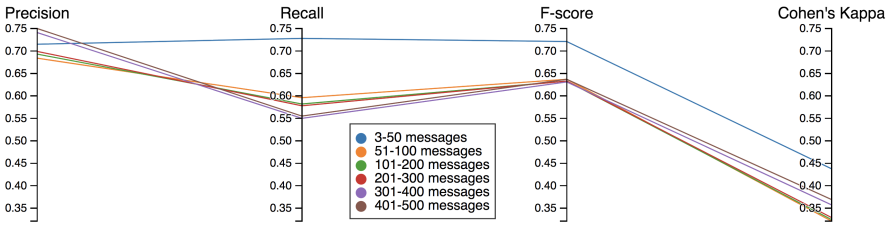


Figure 4: Evaluation results of the Decision Tree model trained using the *distance* feature on different thread-length bins.

MESSAGE LENGTH'S INFLUENCE ON MODEL'S PERFORMANCE Another important aspect is the performance of single features when applied on different length messages. Threads are split into multiple bins, representing threads with an average message length of fewer than 20 words, at least 20 words, at least 40 words and at least 60 words (shown in Table 9).

	< 20	>= 20	>= 40	>= 60
threads	2182	3200	1294	543
"positive"	41,604	74,550	21,869	6,161
"negative"	6,419,375	9,601,351	2,146,466	507,037

Table 9: An overview of datasets having threads with different average message length (in tokens).

The observations show that the performance of the *cosine similarity* (shown in Fig. 5) and the *quote* feature (shown in Fig. 6) are influenced by the average message length. These models perform better on threads having messages with more tokens. Apparently, longer messages have more content information.

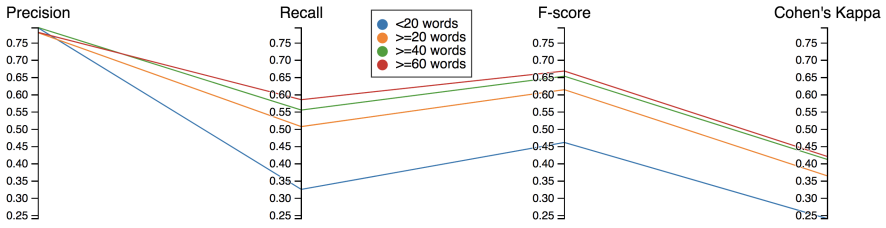


Figure 5: Evaluation results of the Decision Tree model trained using the *cosine similarity* feature on different message-length bins.

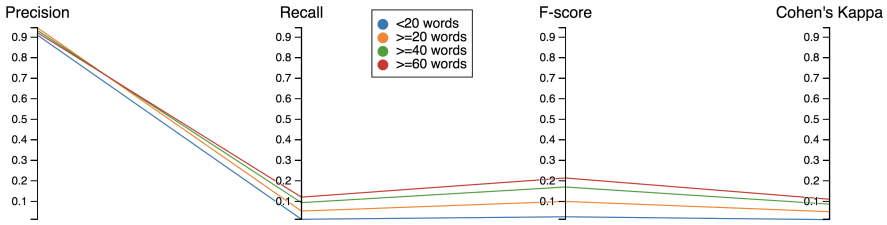


Figure 6: Evaluation results of the Decision Tree model trained using the *quote* feature on different message-length bins.

4.6 ADVANTAGES AND CHALLENGES OF MACHINE LEARNING

The main advantage of machine learning algorithms is in their power to solve complex classification problems. A qualitative training dataset is a prerequisit to train a good and reliable model. Although Chapter 3 lists altogether 17 different features which may describe reply-relations, it is crucial to observe and evaluate these features in order to exclude those which can be misleading.

A trained classifier performs well only on data which has similar characteristics to the training dataset. The characteristics of conversational text data vary though. Thus, for each discussion type, a new (appropriate) model has to be trained.

The performance of the trained model depends on multiple factors. Firstly, presence of reliable features is crucial in order to learn a good model. Otherwise, the model may tend to overfit the training data (as it is previously shown). Secondly, the reconstruction of the reply-relation structure may be seen as an anomaly detection task. Only the minority of all possible reply-relations are correct. Hence, one has to deal with so called imbalanced classes problem. Although multiple methods exist to increase the chance to classify the minority class correctly (e.g., under-sampling), none of them is free of flaws.

All previously described challenges show that although the machine learning is powerful to solve complex classification problems, sometimes it is not enough to evaluate the model using k-fold cross-validation technique to measure its ability to perform a qualitative classification.

5

QUERY BASED MODEL

This chapter describes the unsupervised query based model which can be used to reconstruct the thread structure. Some of the existing work already apply unsupervised rules to extract reply-relations. They, however, use a fixed set of features which is present in their tested dataset. The main advantage of the query based model is its flexibility and interpretability. A subset of features may be selected which describes the parent-child relations best, integrating the user's knowledge on the particular dataset. The results are simply interpretable; it is possible to visually display the influence of single features in the reconstruction's process, and show the certainty of the extracted structure.

5.1 RECONSTRUCTION AS INFORMATION RETRIEVAL TASK

The reconstruction of the reply-relation structure may be seen as an Information Retrieval (IR) task. IR deals with finding documents of an unstructured text that satisfies an information need from within large collections [43]. In the literature, various retrieval models are introduced for different conversational text domains, like emails [56], blogs [20], newsgroups [69], or forums [19]. Retrieval techniques based on a language modeling approach are used to retrieve a relevant post in a discussion or to retrieve a whole thread on a specific topic [52].

The three most used models in IR research are the vector space model, the probabilistic model, and the inference network model. In the vector space model text is represented by a vector of terms, where term gets a non-zero value in the text-vector along the dimension corresponding to the term, if it belongs to the text. The model measures the similarity level between the query vector and the document vector. Probabilistic model estimates the probability of document's relevance for a query. The inference model uses weights for terms in the document, and the document ranking is similar to ranking in the vector space model and the probabilistic model [55].

In a threaded discussion, one reply-relation consists of a parent and a child message. A child may have at most one parent, and it has as many parent candidates, as many messages have been written in the time before it. Query based model uses the idea of an IR model. For each message (except the title message), the task is to retrieve one most suitable parent message from the parent candidate set. The similarity (in this case: the suitability for a parent candidate) is measured, using the presence of features which the user has selected in his query. The query itself is a logical expression; multiple features may be connected using logical operators.

5.2 WORKFLOW OF THE QUERY BASED MODEL

The query is generated in the following way: the user selects a feature subset which he would like to use for the reply-relation reconstruction and creates a logical expression. He can select as many features as desired.

For each feature, he can define the minimum similarity level (for content features), maximum distance between messages (for structural and meta data features). He can weight each feature, specifying its importance in the decision making process. An example of a query is shown in Fig. 7

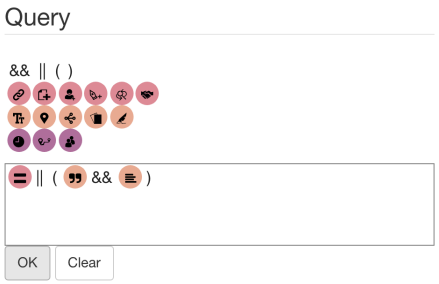


Figure 7: An example of a query (*cosine similarity* || (*quote* && *substring*)).

When the query is executed, the regular expression is sent to the server. There, for each possible reply relation, the system inquiries, if this relation satisfies the given query. All valid relations are seen as relation candidates. Then, the system verifies, if for one single message multiple parent candidates exist. In such a situation, the highest relation score within the group of candidates is calculated. It means that for each feature in the query the system compares the similarity values for content features, distances of meta data features, and the values of structural features. The final relation score is the weighted sum of comparison of all features.

Parent candidate extraction

For each message, a list of parent candidates is stored, sorted to their scores. The first message (having the highest score) is seen as the most suitable parent message.

Frequently, the system may detect multiple parent candidates for a single message, all satisfying the given query. The decision which parent is the most suitable is not trivial, in some situations the system identifies the correct parent message as a parent candidate but selects another candidate as the most suitable one. It is important to maintain and provide such information to the user, as it means, that the system has made the wrong choice only in the last decision making step.

Candidates with equal scores

If multiple relation candidates for one single message have the same score, then the first relation (in temporal order) is chosen as the most suitable one. This is a heuristic, which might be changed. Some authors use the closest previous message as the default parent message. It is not a reliable heuristic for the Reddit data though, as the previous message is the actual parent only for 5% relations in average.

5.3 MODEL'S PERFORMANCE

The same representative sample of 40 Reddit discussions, used to evaluate the machine learning model, is used to explore the performance of the query based model.

PERFORMANCE OF SINGLE FEATURES The model is applied on a single feature at a time, using default parameters. Three metrics are used - precision, recall and F-score, to describe model's performance (shown in Table 10).

Feature	Precision	Recall	F-score
Cosine Similarity	**0.37**	**0.12**	**0.18**
Url Enrichment	**0.35**	**0.12**	**0.18**
Topic Enrichment	0.17	**0.12**	0.14
Author Enrichment	0.09	0.04	0.06
Wordnet Enrichment	**0.30**	**0.12**	**0.17**
Word Embedding	**0.37**	**0.11**	**0.17**
Topic Agreement	0.09	0.05	0.06
Quote	**0.80**	0.04	0.08
Substring	**0.58**	0.04	0.08
N-Gram	**0.21**	0.04	0.06
Named-Entity	**0.24**	0.07	0.11
Lexical Episode	0.15	0.03	0.04
Coreference	0.15	**0.11**	0.13
Reference to Author's Name	1	0.01	0.01
Time-Distance	**0.21**	**0.21**	**0.21**
Distance	0.05	0.05	0.05
Different Authors	**0.20**	**0.20**	**0.20**

Table 10: Features are tested on 40 Reddit discussions. The precision, recall and F-score show the average performance of the query based model using only one feature as an input query.

Content features such as *cosine similarity* or different message enrichment techniques have relatively high precision. Their performance depends on the selected minimum similarity-threshold. By increasing this value, the precision can be improved (but at the cost of recall). The query based model reaches the highest precision on the tested data, when *quotes* or *substrings* are used as queries. Using *quotes*, the average precision of the model is 0.8. However, the recall is only 0.04. Thus, the *quote* is a reliable feature, but it alone can reconstruct only a small part of reply-relations existing in the Reddit data. One needs to take into account, that features like *quote*, *substring*, or *cosine similarity* frequently overlap. To increase the recall, one can use more frequent features such as *coreference* or *topic agreement*, or decrease the similarity threshold for content features, but in such case more false reply-relations would be detected.

PERFORMANCE OF FEATURE COMBINATION Each single feature used separately may not reconstruct the whole reply structure, as usually, not all messages linked with a reply-relation have a high content similarity, mention the same named-entity, or use quotations. In order to reach higher model's performance, multiple features have to be combined. To evaluate the query based model, we use the previously mentioned

sample of 40 Reddit discussions. The following query is executed: "((*quote* (weight: 5) || *substring* (weight: 4) || *cosine similarity* (min similarity: 0.2, weight: 3) && *different author*)||*time-distance* (max-distance: 24 hours, weight: 1)". The evaluation results are shown in Table 11.

Files	Precision	Recall	F-Score
"Politics (60-120 msgs)"	0.39	0.30	0.34
"Politics (160-210 msgs)"	0.36	0.30	0.33
"World News (60-120 msgs)"	0.40	0.30	0.34
"World News (160-210 msgs)"	0.29	0.25	0.27
Avg.	**0.36**	**0.29**	**0.32**

Table 11: Evaluation results of the query based model (using a query: "((*quote* (weight: 5) || *substring* (weight: 4) || *cosine similarity* (min similarity: 0.2, weight: 3) && *different author*)||*time-distance* (max-distance: 24 hours, weight: 1)").

Although the results seem to be modest, the F-score of the query based model is 24% higher than of the trained classifier (presented in previous Chapter).

In Chapter 2 we show that most of the related work use relatively short threads to train and test the model. Thus, we crop the 40 test discussions by extracting the first 30 messages in a temporal order from each of them. These artificially created 30-message long discussions are tested using the same query as previously described. Results are listed in Table 12. Apparently, both, precision and recall, are improved.

Files	Precision	Recall	F-Score
"Politics (60-120 msgs)"	0.60	0.47	0.52
"Politics (160-210 msgs)"	0.63	0.57	0.60
"World News (60-120 msgs)"	0.48	0.40	0.43
"World News (160-210 msgs)"	0.51	0.46	0.48
Avg.	**0.56**	**0.48**	**0.51**

Table 12: Evaluation results of the query based model applied on 30 message long threads.

By decreasing the thread size to 10 messages, the overall precision, recall and F-score of the model increases even more (shown in Table 13). These results emphasize that the performance of the model depends on the thread size. Thus, on short discussions, the query based model is able to compete with already existing reconstruction methods.

Files	Precision	Recall	F-Score
"Politics (60-120 msgs)"	0.78	0.70	0.73
"Politics (160-210 msgs)"	0.84	0.81	0.83
"World News (60-120 msgs)"	0.55	0.54	0.55
"World News (160-210 msgs)"	0.63	0.61	0.61
Avg.	**0.70**	**0.66**	**0.68**

Table 13: Evaluation results of the query based model applied on 10 message long threads.

5.4 ADVANTAGES AND CHALLENGES OF THE QUERY BASED MODEL

The reconstruction of the reply-relation structure is challenging, especially on relatively long threads having infrequent reliable features. Thus, it is important to provide information how certain the extracted relations are and why some reply-relations may not be reconstructed at all.

FLEXIBILITY As it is shown in Table 10, some features are highly infrequent or even missing in the tested Reddit data. The system should be adaptable for various conversational texts. Thus, these features are maintained in the feature set, as they could be present in other discussion datasets.

One advantage of the query based model is that its performance is not negatively influenced by the infrequent features. The user can exclude a feature from the query which is not present in the data. Thus, the model is applicable on various input datasets with different message characteristics. That is also the main difference to the supervised learning algorithms, where the system learns correlations of features existing in the training data.

SIMPLE TO EXTEND The query based model is simple to extend. As it does not learn the model but reconstructs the thread structure based on the queries, it is simple to add a new feature to the feature set. One possible extension could be a question-answer pair detector. As almost 10 percent of messages in the tested Reddit discussions end with a question mark, it might increase the model's performance, if question-answer pairs were detected.

CHALLENGES OF THE MODEL Despite the simplicity and interpretability of the model, there are still some challenges to deal with. First of all, the reliability of the features depends on the input dataset. Having the possibility to change the similarity threshold, distance and weighting parameters, the user may influence results based on his knowledge on the used dataset. But it is true only if he is aware of the feature performance. To get this knowledge, he would need to explore the distribution of single features with different similarity/distance parameters, which is inefficient to do. Thus, it is important to provide such information by the system.

6 VISUALIZATIONS

This chapter shows the visualization and interaction techniques which we use to present the reconstructed reply-relation structure. There are two goals which the used visualizations should satisfy. Firstly, they should help the user understand why particular relations are extracted. Presence of multiple features in a message-pair may indicate an existence of a reply-relation. However, multiple parent candidates for a single message may point out that the computed reply-relation might be detected wrong. A visual representation of such evidence can be useful, either the dataset has a ground truth structure, or not. If the ground truth structure is given, the user can use visualizations to learn the usefulness of different queries for similar data, or to compare different models to each other. However, if a dataset has no given structure, the evidence on the feature presence and the reply-relation certainty is crucial for building a trust to the reconstructed structure. Secondly, the visualizations should provide a clear overview of the discussed subtopics. All visualizations are created using JavaScript, HTML and D3 library.

The system has two main visualization components: the *overview* showing the whole discussion and supporting the *close reading* [30], and the *forest view* which can be disentangled in two ways. Firstly, a *parent-child space view* gives a better insight into the computed reply-relation certainty and provides an overview of similar messages. Secondly, using interaction techniques, the complex structure is split into given or computed connected components (displayed in the *disentangled forest view*), representing discussion's subtopics.

All visualizations have something in common; the title message, being displayed on top of the view, is treated extraordinary. It is not influenced by filtering or sorting functions (introduced in following sections), due to its special meaning. This message is a discussion's root; it specifies the thread's topic, and always has at least one child message. Thus, it is always present in all visualizations.

6.1 VISUAL ELEMENTS

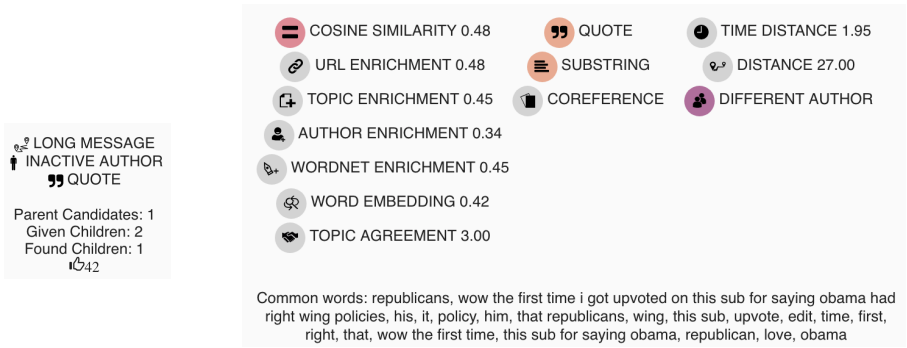
We use the idea of Thread Arcs [33] visualization to represent the given and computed reply-relation structure. The basic visual elements are nodes and links between them. One reply-relation consists of two nodes (representing two messages), and a connecting link.

To display a single message, we use a circle (●) which radius is scaled to message's word count. There are situations when one message can have multiple parent candidates. Only one relation (having the highest score in comparison to other relation candidates for this particular child), is displayed in the default *forest view*, other candidates are only shown on demand or in the *parent-child space*. The certainty of this

particular (highest scored) reply-relation is lower than for messages with only one parent candidate. This information is encoded in the borders of the child node (or). The crispier they are, the more uncertain is the relation to its parent. Circles of *junk* messages are colored black; all other message circles are displayed gray. If the given or found parent of the message is the title message, then a small white circle is displayed on top of the gray/black node ( ).

Reply-relations are encoded using links between messages. If the dataset has a given reply-relation structure, the color is used to show which relations have been identified correctly (**green**). All other relations are displayed gray.

The tooltip for a node (message) shows which categories it represents, the statistics of its computed / given children, and the up-votes (Fig. 8(a)). The tooltip for a link (reply-relation) shows, which features influence its classification (Fig. 8(b)). All features present in the relation are displayed, for numerical features the computed value is shown. Colored features are contained in the currently executed query.



(a) Tooltip of the node (message).

(b) Tooltip of the link (reply-relation).

Figure 8: Tooltips are used to display the information on present reply-relation features or message categories.

6.2 OVERVIEW

In the *overview* (shown in Fig. 9), all messages of the discussion are displayed in the temporal order. Messages classified as *junk* are highlighted using darker borders. This view can be used for *close reading*.

If the data has a given ground truth structure, then this structure is displayed on the left side of the view. There, the thread arcs representation is used, to show an overview of the given reply-relation structure. Message pairs which represent reply-relations are connected with a link. The score (up-votes) of each message is shown as a glyph ( ), if given in the data. It gives an overview of the most up-voted messages, which could also be seen as the most important messages in the discussion.

On the right side of the view, the currently computed reply-relation structure is shown; it is represented the same way as the given structure (with nodes and connecting links). On this side, however, the glyphs represent the number of children for

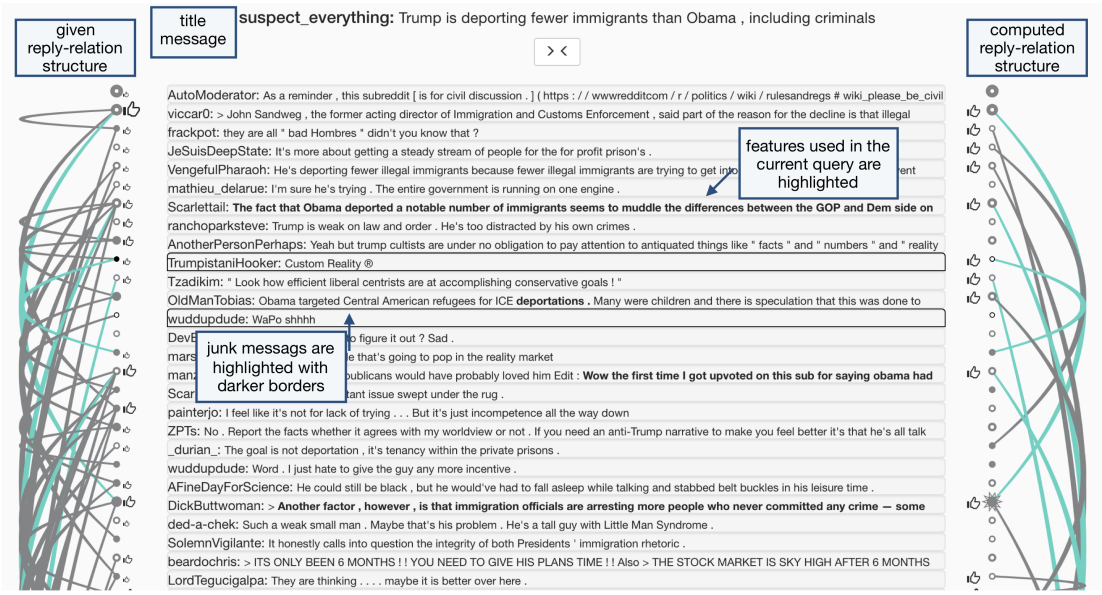


Figure 9: Overview.

the particular message, due to a high correlation between up-votes and the number of children. That allows to reconstruct possible up-votes for a dataset with no given ground truth structure.

INTERACTIVITY By clicking on a message, its parent and children are highlighted, others messages are being faded out (shown in Fig. 10). It allows the user to explore a smaller part of the discussion easier.

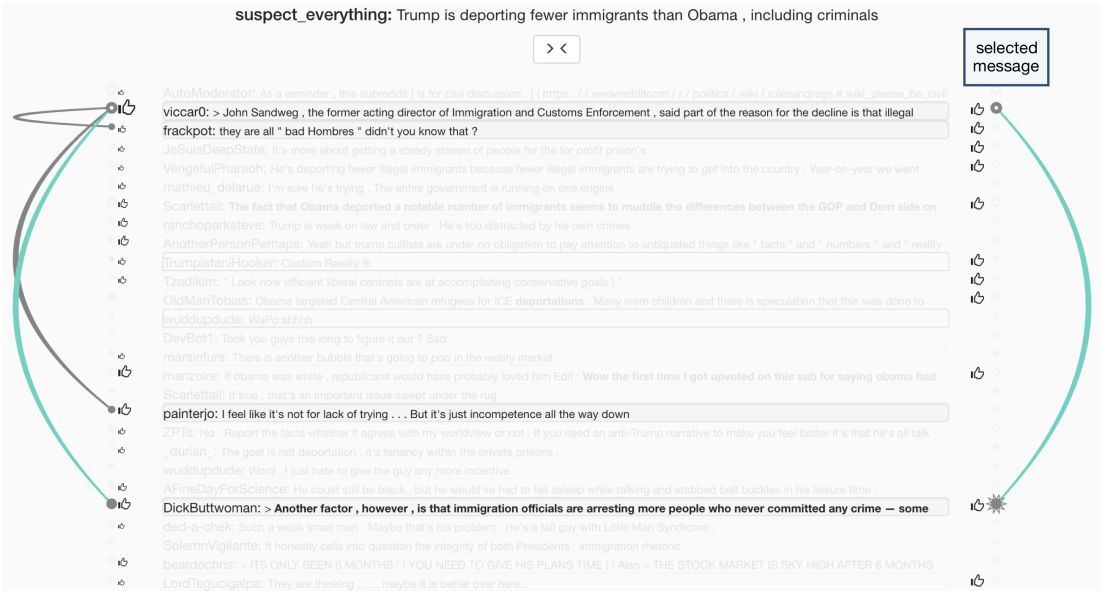


Figure 10: By clicking on a message, its parent and children are highlighted.

By hovering over a message, its content is fully displayed for a *close reading* (shown in Fig 11).

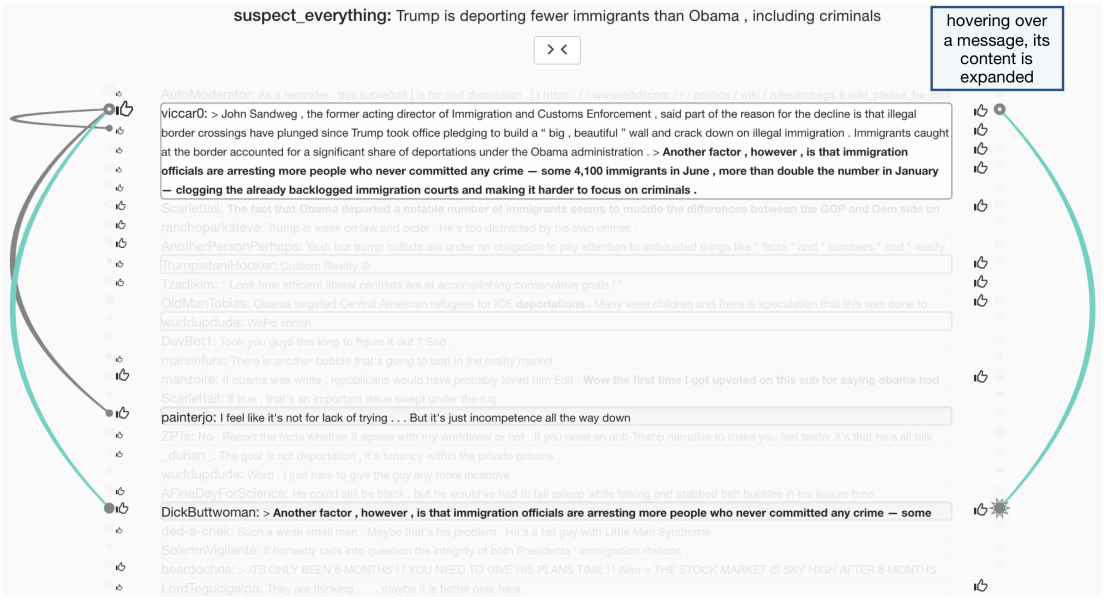


Figure 11: By hovering over a message, its content is fully displayed for a *close reading*.

LIMITATIONS The visualization could be improved, allowing the user to select only a subset of messages of his interest. For example, a selection of messages having most up-votes or most children, could provide a summary of the discussion.

6.3 FOREST VIEW

To learn more about the extracted reply-relation structure, its certainty and the discussion's subtopics, the user may explore the *forest view* (shown in Fig. 12), by clicking on the button under the discussion's title. In this view, messages are hidden and displayed only on demand in the *close reading view* placed on the left side of the view (the structural features which are currently selected in the query are highlighted). The thread arcs structure of the given and computed relations remain similar like in the *overview*, being moved together in the middle of the view. Here, the user may zoom out the view to observe the complete discussion's structure at once.

The number of features which the reply-relation contains, satisfying the selected settings, is encoded in the width of the link's curve. This width can be an indicator, which reply-relations are more probable to reconstruct correctly. Reply-relations, where none of the features are part of the used query, are partly faded out using a gradient function.

The *forest view* can be expanded in two ways. Firstly, the *parent-child space* gives an overview of the certainty of the extracted reply-relation structure, showing the distribution of the parent candidate set for each message. Secondly, the *disentangled forest view* represents existing subtopics in the data.

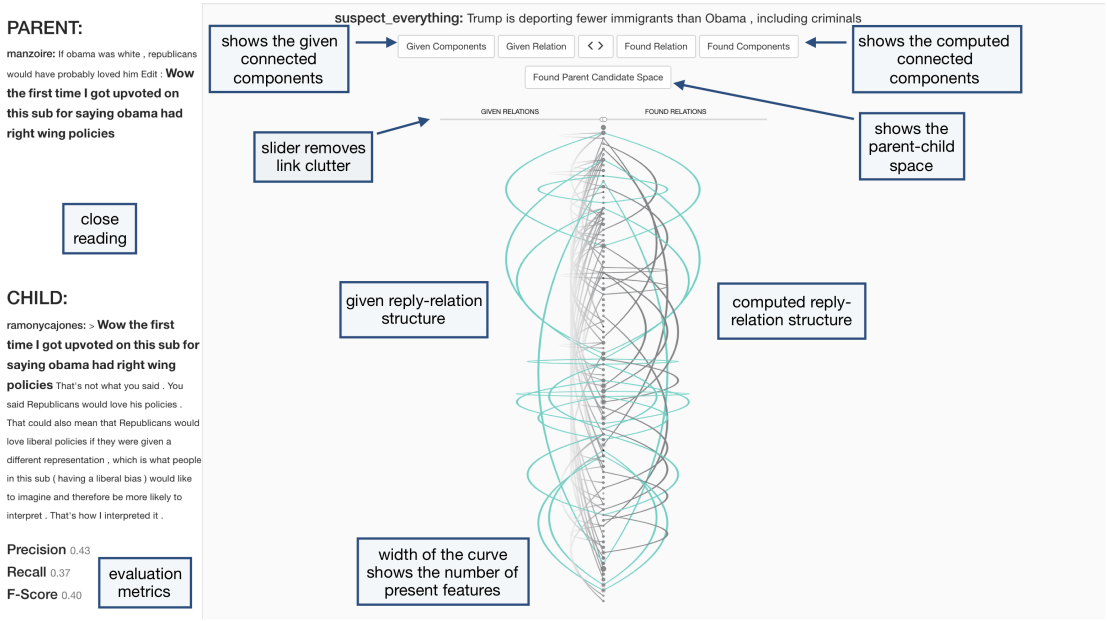


Figure 12: Forest view.

6.3.1 *Parent-Child Space*

The *parent-child space* visualization gives an overview of the certainty level of computed reply-relations. The certainty is described by the number of extracted parent candidates; the more parent candidates for a single message exist, the more probable is that the system has selected the wrong parent as the most suitable one. An example of the *parent-child space* is shown in Fig. 13.

The idea behind this visualization is that "the child messages visit their parent candidates, beginning with the most certain ones". On the left side of the view, the computed structure is shown using the basic Thread Arcs technique. For messages having at least one parent candidate, the distribution of their parent candidate "visits" is displayed on the right side of the view. There, child messages (shown as circles) are positioned at the height of their parent candidates. Visiting each next (less probable) parent candidate, the child's x position is moved 300 pixels to the right side. By linking the same child over its "visits", one can get a good overview of the reliability of the used model. The more long paths are generated, the less descriptive is the used model for the thread structure's reconstruction task.

If multiple child messages have the same parent candidate (which has the same index in the parent candidate lists), then they are displayed in a row next to each other. This representation groups similar messages existing in the discussion. The more messages are displayed close to each other, the more representative they are for summarizing the discussion's content.

Child messages "visiting" a parent candidate, which is also their parent in the ground truth data, are colored **green**. All child paths till this "visit" are colored **yellow**. These visual variables are present only if the data has a ground truth information.

This representation is helpful for a comparison of multiple models to each other. A model where more **yellow** paths are displayed than the gray ones, can be seen as more descriptive.

The view provides an overview of features present in the computed relations. They are displayed between the Thread Arcs structure and the child-paths. Features used in the current query are colored, others are being displayed gray. Relations where only few features are present can be seen as less reliable.

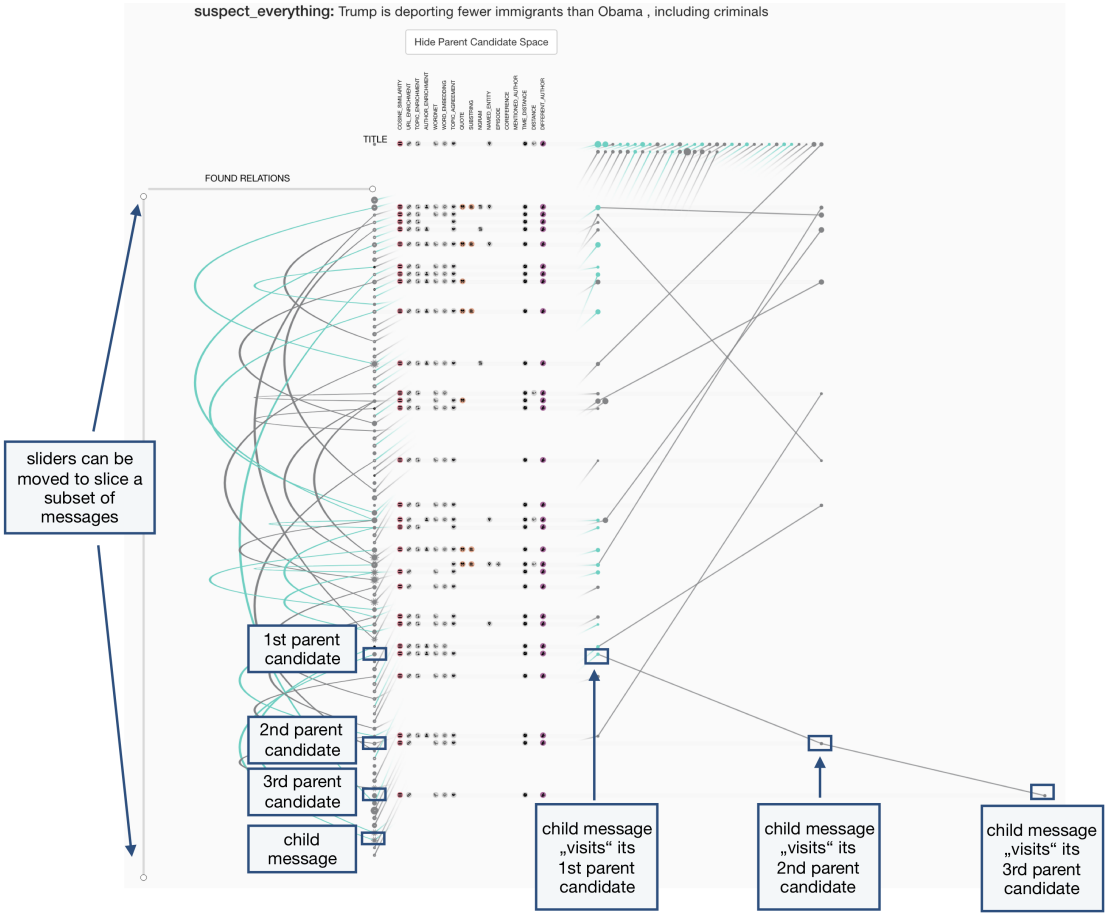


Figure 13: Parent-child space.

INTERACTIVITY The *slice and dice* technique is used to reduce the amount of displayed data. By moving a slider, the user can specify a subset of messages which he would like to explore.

This interaction technique is usable not only to reduce the link clutter, but also to temporally exclude messages from the analysis. The system updates all parent candidates regarding the slider's position. Thus, it allows the user to analyze discussion's subsets independently. The user can, first, sort messages to a specific attribute (e.g., message length), and work on different message groups separately. An example of the *slice and dice* technique is shown in Fig. 14.

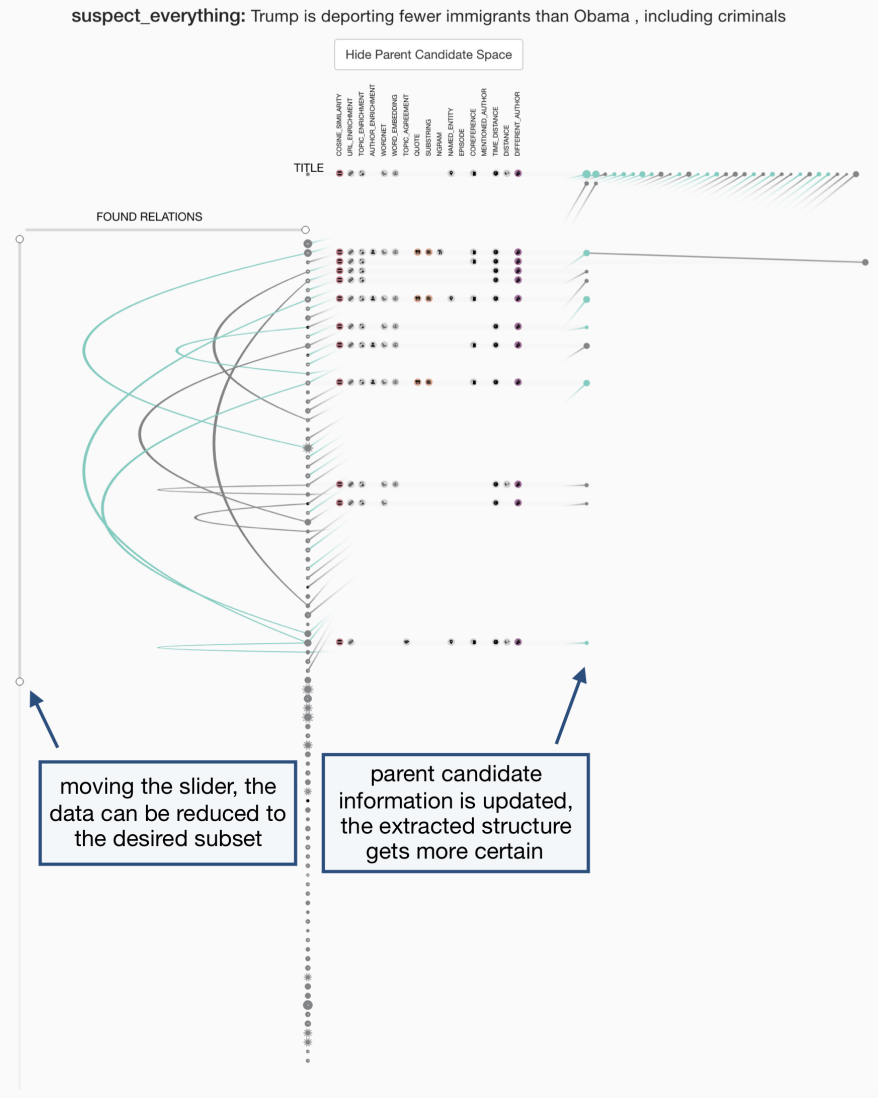


Figure 14: *Slice and dice* technique. A subset of messages can be selected for further analysis.

For data without ground truth structure, the *slice and dice* technique can be used in a combination with a *brushing* technique. We use *brushing* to allow the user to select child messages, which are assumed to be correctly linked to their parents (shown in Fig. 15). Selected messages are linked to their parent candidates using the particular y position of the child. These relations are set fixed, and are not influenced by the successive execution of other models. Thus, the reconstruction of reply-relations may be done in multiple steps. Fixed relations are colored in another shade of **green**.

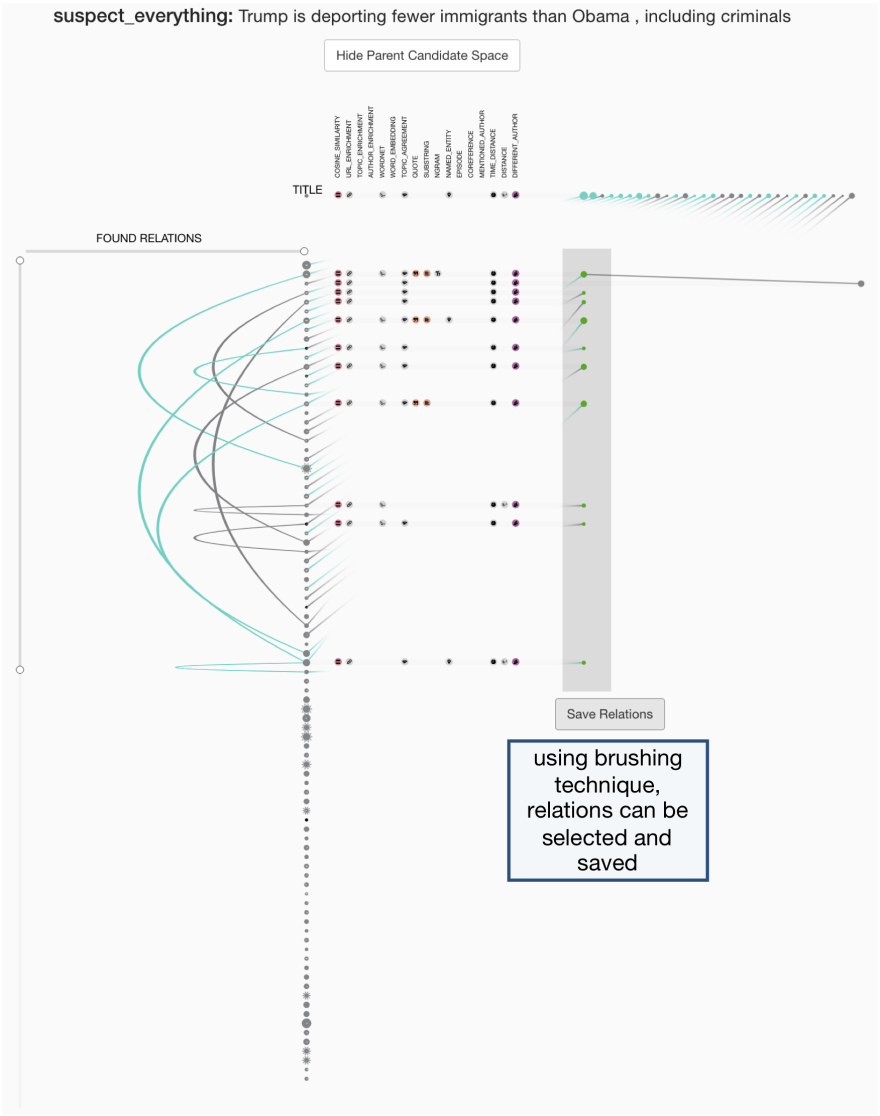


Figure 15: *Brushing* technique. Selected relation subset is stored and not influenced by the successive execution of other models.

By hovering over a row, the content of the parent and all its child candidate messages is displayed in the *close reading view*. By hovering over a child's path, the content of the child and all its parent candidates is updated in the *close reading view* (shown in Fig. 16). Additionally, the path is highlighted, enabling a better overview of the linked messages. All features, which are present in the current query, or which are used to train the selected classifier, and which are existing in the particular relation with a score greater than 0, are displayed on top of the child circles. That gives a good overview of features which influence the relation classification. Using a trained classifier, the score of the classified reply-relation in displayed next to the feature icon.

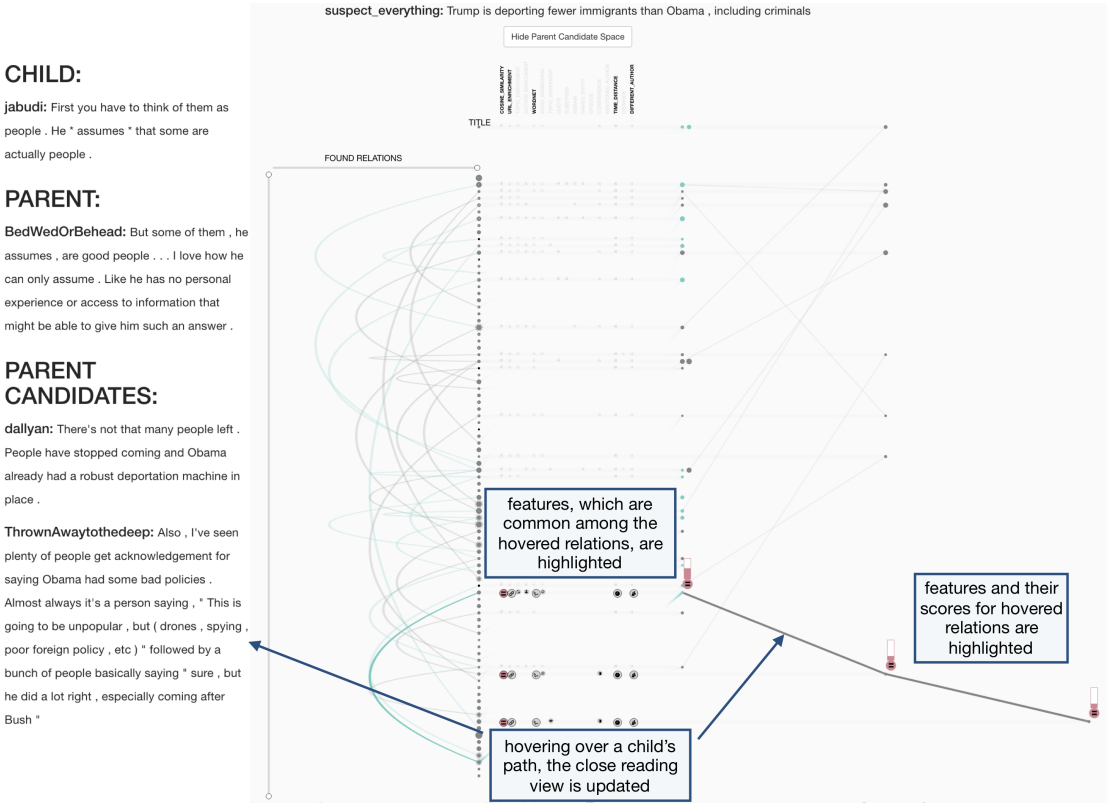


Figure 16: By hovering over a child's path, the *close reading view* is updated. All features in hovered relations are highlighted.

LIMITATIONS The representation of the model's reliability can be improved. Currently, the **green** and **yellow** colors are used to highlight child-paths, where the correct parent is among the parent candidates. It might be useful to visually represent if the transitive reply-relations are given in the ground truth structure. If the system detects a message as a parent candidate, but in the ground truth the child or the parent of the detected message is the searched parent, it may indicate, that the model's reconstructed structure is relatively close to the ground truth.

6.3.2 *Disentangled Forest View*

The second expanded representation of the *forest view* is the *disentangled forest view*. There, transition techniques are used to disentangle the discussion's structure. By clicking on one of the side buttons under the title message, the given or computed reply-relation structure is shown accordingly. Messages, which reply to the title and have no children, are joined in one separate connected component.

During the transition, one compact *forest view* is split into multiple connected components. These components can be moved vertically closer together, losing their temporal information, and showing more compact structure. They can be sorted to their size (number of messages), giving an overview of the largest discussion's subtopics. An example of the *disentangled forest view* is shown in Fig. 17.

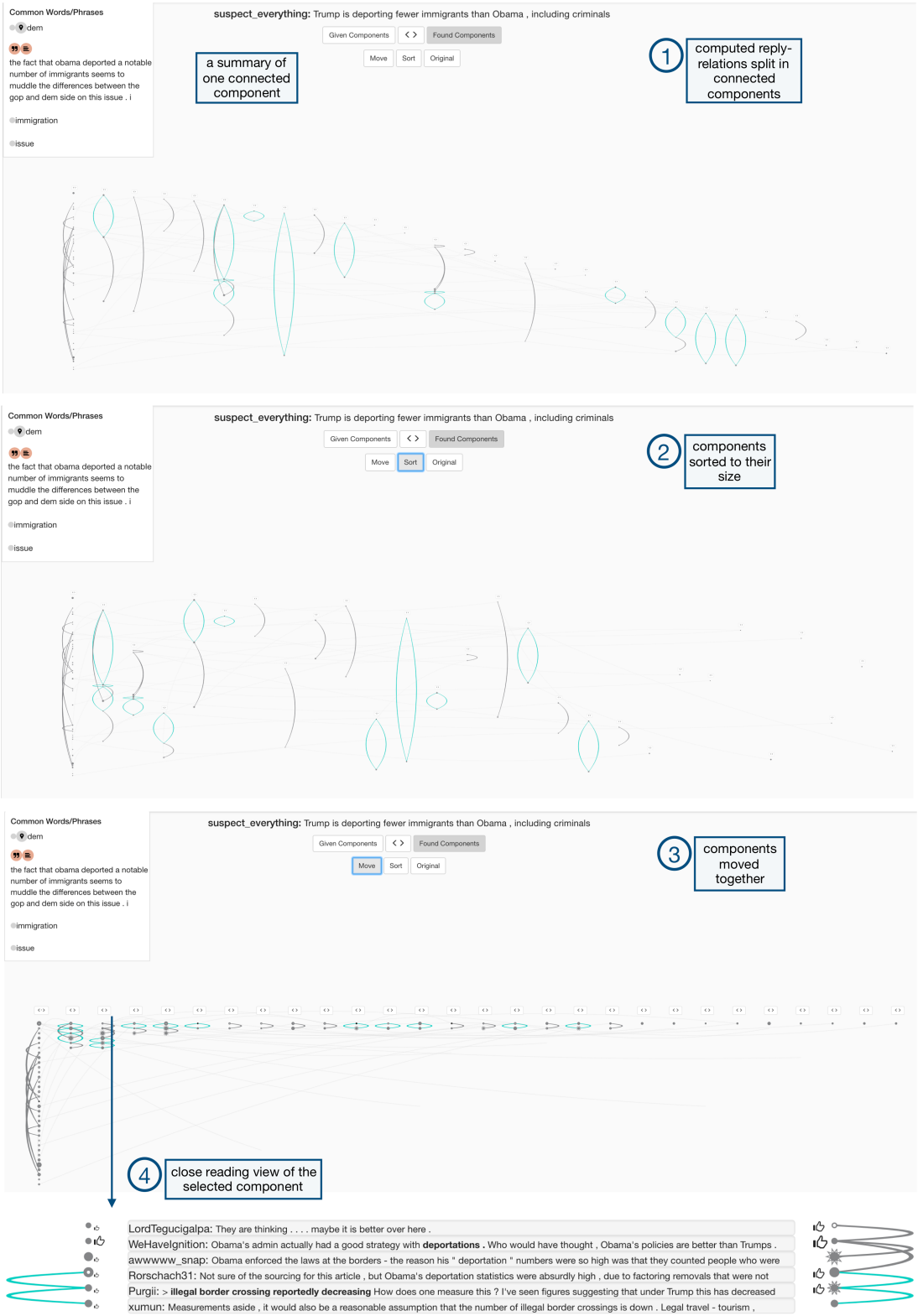


Figure 17: Disentangled forest view.

INTERACTIVITY There is a reason, why messages are part of one connected component - most frequently they share common words, as they discuss similar subtopics. By hovering over one connected component, its common words and common phrases are displayed next to the *close reading view*, summarizing the particular subtopic. Each word or phrase includes an icon of the feature it represents (e.g., a ❞ representing a *quote* phrase).

By clicking on the button displayed on top of the connected component, an *overview* visualization is opened in a separate view, showing messages of the selected connected component.

LIMITATIONS The *disentangled forest view* can be used to compare the given and the computed connected components to each other (if the used data has a ground truth structure), or to observe the computed discussion's subtopics in the dataset without the ground truth information. Regarding the first use case, two *disentangled forests* (given and computed) could be displayed simultaneously, highlighting components which have some reply-relation matches. Such representation could give an additional insight, which subtopics are simpler to reconstruct (e.g., due to the presence of content information).

6.4 GENERAL INTERACTION TECHNIQUES

We use the "information-seeking mantra" of Ben Shneiderman [54] for the *forest-view* visualizations. There, first an overview of the whole reply-relation structure is shown. On demand, the system provides more detailed information on the relation certainty, and discussed subtopics. In addition, filtering and sorting functions can be applied to reduce the amount of content-free messages, or display patterns regarding the selected sorting attribute.

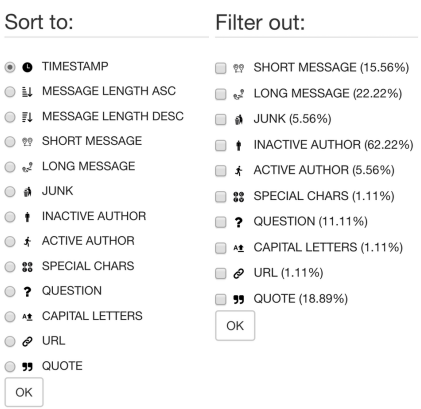


Figure 18: Messages can be sorted by multiple attributes. User can filter out a message category, if needed.

SORTING FUNCTION By default, messages in the *forest view* are displayed in temporal order. When messages are sorted by another attribute (e.g., message length), some additional patterns might be detected. Sorting function is provided in the settings sidebar.

FILTERING-OUT MESSAGE CATEGORIES As not all of the message relation categories may be extracted reliably, sometimes it is useful to filter out messages, for which reply-relations most frequently can not be classified correctly (e.g., *junk* messages). In the settings sidebar the user can select message categories, which should be excluded from the thread, then only messages representing the remaining categories are displayed. Different sorting and filtering attributes are shown in Fig. 18.

HIGHLIGHTING OF FEATURES User may explore the dataset and present features in the given or computed reply-relation structure. By clicking on one feature in the settings sidebar, the relations containing this particular feature are being highlighted, and the rest of the relations are faded out. It allows the user to manually explore the data, depending on his interests.

UPDATING CLOSE READING VIEW By hovering over a single message, its content is shown in the *close reading view*. Besides, by hovering over a link, the content of the parent and the child message is updated.

SANDBOXES In situations, when the dataset contains the ground-truth structure, the system should show as many clues as possible, which relations can be reconstructed reliably. Therefore, we use a concept of *sandboxes* (shown in Fig. 19). It is a small representation of parent-child relations, where a summary of message categories describes the correctly and incorrectly reconstructed reply-relations.

To display the *sandbox*, the user drags a message circle to one side. Then dragged circle and its children are displayed in a separate modal view, giving a short summary on affected message categories.

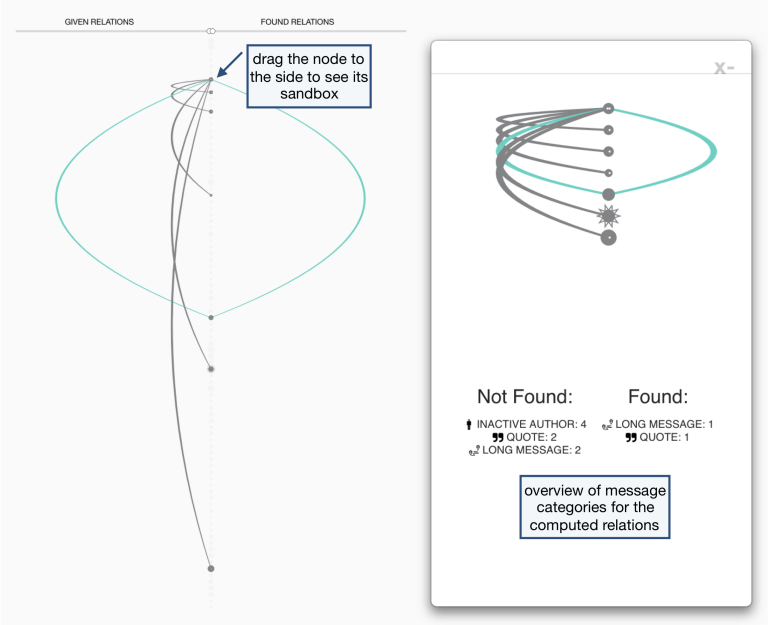


Figure 19: Sandboxes are used to show an overview of relations between a parent message and its children.

7

VISUAL ANALYTICS FRAMEWORK

This chapter describes the visual analytics framework which supports the reconstruction of the reply-relation structure. It incorporates both methods - the presented query based model (Chapter 5) and a learned classifier; these two models may be applied separately or in a combination. The framework is flexible to be used on diverse conversational texts having varying characteristics. It provides information on the present features and allows the user to integrate his knowledge in the reconstruction process. The system provides visual evidence on reconstructed reply-relation certainty, and entangles the complex structure, delivering more insights into discussion's content.

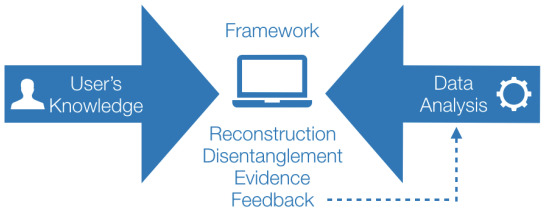


Figure 20: Framework uses the user's knowledge and automatic data analysis methods to reconstruct the reply-relation structure, disentangles the discussion's content, provides an evidence on structure's certainty and considers the user's feedback in the next reconstruction's cycle.

Complex real-world data cannot be efficiently and properly analyzed using automatic computational methods or interactive visualizations separately. It is important to combine these techniques to reach the highest profit. Keim et al. [32] state that visual analytics provides technology that combines the strengths of human and automatic data processing; it strives at multiplying the analytical power by finding effective ways to combine interactive visual techniques with algorithms for computational data analysis. Hence, we use the advantage of visual analytics and provide a framework (Fig. 20) to fulfill the following tasks:

- The extraction of the reply-relation structure should take user's input, and system's suggestions into account.
- A possibility to combine multiple models should guarantee a level of flexibility.
- The tool should provide visual evidence, how reliable and certain the extracted structure is, and which features influence the results most.
- The reconstructed structure should be stored for further usage and analysis.

7.1 USER'S INPUT AND SYSTEM'S SUGGESTIONS

Stolper et al. [60] write that the user-driven analytics are statistical algorithms which enable analysts to incorporate their domain knowledge or data-driven insights into the analysis. For the reply-relation reconstruction task, it is important to integrate user's knowledge of the used dataset if it is given. Some background information like a higher likelihood that the child answers to the previous message rather than to the title message can improve the model's performance.

Like it is shown in Chapter 2, most of the models presented in related work are applicable only on one specific dataset having a particular set of features present. Thus, for different datasets, different models have to be applied. In order to choose the best model, the user has to be aware of the characteristics of the used dataset, or he has to spend time on exploring the data. Hence, it is important that the system provides suggestions, which features might describe reply-relations for the used dataset best.

Feature frequency

The system evaluates all features, and provides their frequency in computed reply-relation candidates. The frequency is shown as a tooltip for the particular feature and is displayed in the settings sidebar. It is shown as a percentage, where 100% indicate that there are at most n (number of messages in the input discussion) reply-relations having this feature with currently selected similarity or distance thresholds. Features, which are present in more than 100% relations, are less descriptive for the *RTS* task.

Default similarity thresholds

Selected similarity thresholds for content features highly influence classification results. A manual selection of the most suitable parameters for the input data may be time consuming. Thus, the system provides default similarity thresholds, being most representative for the input data. These thresholds encode the highest similarity values for which not more than n (number of messages in the input discussion) computed reply-relations having the particular feature are detected.

The system analyzes message categories present in the data, and provides an overview of their frequency. Thus, the user is aware of the quality of the used dataset and can exclude some less descriptive messages from the dataset if needed (e.g., *junk*).

7.2 EXTRACTION OF REPLY-RELATION STRUCTURE: PIPELINE

Multiple possibilities exist how to reconstruct the logical thread structure. The user can use a trained classifier, he can create a manual query, apply a heuristic, or combine some of these steps. A pipeline of the executable steps is shown in Fig. 21.

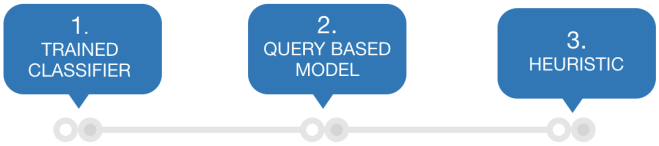


Figure 21: Pipeline has three steps: a trained classifier, the query based model and the heuristic. User can select each step separately, or use multiple steps in the given order.

These single steps are applied in a sequence; the subsequent step uses the input (valid reply-relation candidates) generated by the previous step. Thus, if all steps are executed, the classifier is used to extract a set of reply-relation candidates first. This candidate set is used as an input for the query based model. There, the system inquiries, which reply-relation candidates satisfy the given query and extracts at most one suitable parent for each child message (maintaining the information about parent candidates). Afterwards, the heuristic is applied; each child message having no valid parent is linked to the title message.

USING TRAINED CLASSIFIER The user can apply a trained classifier to extract the reply-relation structure. Multiple models are provided for selection. Like described in Chapter 4, the current models trained on the Reddit data have a poor precision and can not reconstruct the structure completely. But, they can be used as a preprocessing step for the query based model. The trained classifier can dismiss part of "negative" relations from the reply-relation candidate set. This can improve the query based model's performance on less reliable features (e.g., *named entity*, *topic agreement*).

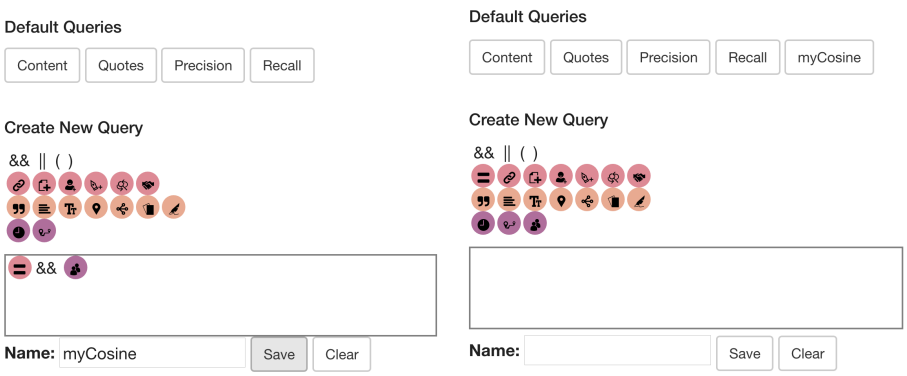
A list of classifiers is provided in the settings-sidebar. There, the user can try out multiple learned classifiers (Decision Tree model or Random Forest) trained on 5 features (presented by Aumayr et al. [3]), or the same models trained on 13 features, described in Chapter 4.

USING MANUAL QUERY The query based model has multiple advantages against the trained classifier. First of all, it does not overfit. The user can create different queries, and the system provides an evidence how reliable the extracted relations are, by showing the presence of features and the number of possible parent candidates for each of the child message. The user can use this knowledge to adapt the query and improve the certainty level of the extracted reply-relations. The query based model is unsupervised. Hence, it can be applied on discussions where no ground truth information exists.

The query can be used to enrich the system with the user's knowledge. If the user is aware, that it is likely that the messages are linked to the previous messages in the temporal order, he can use a query containing the *distance* feature with maximum *distance* 1. This feature can be weighted lower than other features (e.g., *cosine similarity*), ensuring, that the *distance* feature will only be applied for those messages where no parent is found.

The system provides a few default queries, which are created based on our observations on the feature reliability present in the Reddit data. These queries should be replaced by automatically generated ones, based on the analysis of the input dataset.

The user is able to manually add queries to the default query list, like it is shown in Fig. 22. Queries which perform well on the used dataset can be simply reused, improving the usability of the tool.



(a) The user can create and save a new query.

(b) The saved query is added to default queries.

Figure 22: The user can manually save the best performing queries.

Currently, the query is located in the sidebar. Using a drag and drop method, the user can select single features and logical operators. The drag and drop method could be replaced by a click, saving the time spent on the creation of the query. Also, the usability of the query should be improved, to let the user change the order of single elements if desired.

USING HEURISTIC Sometimes it is likely that the most frequent parent message is the discussion's title. By selecting the "Use Heuristic" checkbox in the settings sidebar, the system will assign the title message as parent to all messages, where no appropriate parent message has been found after the execution of the classifier or query. This heuristic is expanded by the idea of [4] that an author usually does not reply to the title post in his/her second message. Thus, the system proves if the message is the first of the particular author, only then the title message is assigned as the most suitable parent.

The pipeline is displayed in the sidebar; different steps and their usage are shown in Fig. 23.

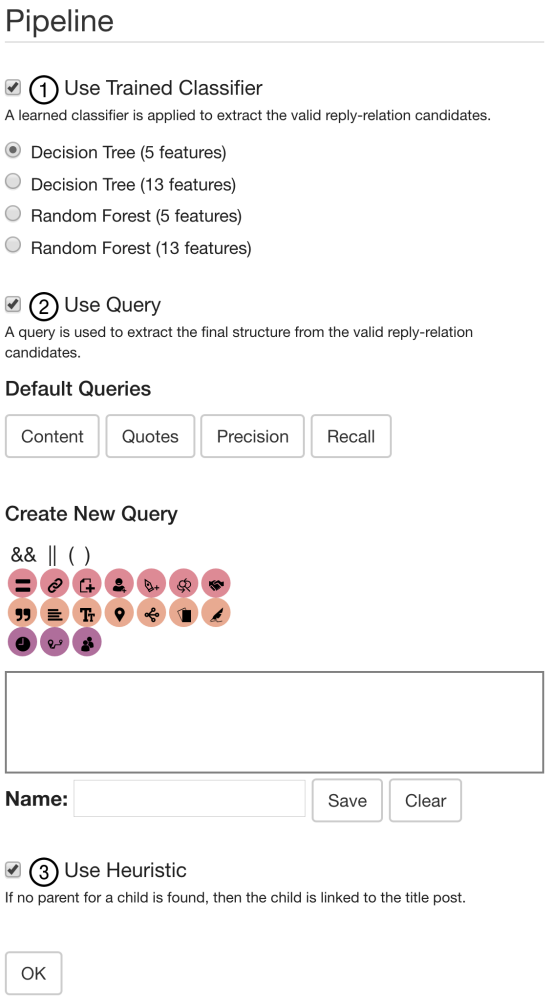


Figure 23: Pipeline and its steps are displayed in the sidebar.

7.3 VISUAL EVIDENCE ON THE RESULT CERTAINTY AND ITERATIVE RECONSTRUCTION

Interpretability of a classification model is emphasized by multiple authors, like in [63, 21]. Freitas A. A. [21] writes that the importance of comprehensible classification models stems from several issues. Understanding a computer-induced model is often a prerequisite for users to trust the model's predictions and follow the recommendations associated with those predictions. The need for comprehensible models is also strengthened when the system produces an unexpected model; then the user requires an explanation from the system as a requirement for model acceptance. In some application domains, users need to understand the system's recommendations to explain the reason for their decisions to other people.

When the reply-relations are reconstructed, the *parent-child space* visualization provides insights into the structure's certainty. If a message contains only one parent candidate, it is certain. If it has multiple parent candidates, then the certainty level

decreases. In such situations, there is a higher chance that the system has selected the wrong parent message as the most suitable one.

In Chapter 3 we present that the quality of individual messages varies. Thus, the reconstruction of different message categories may have different relevance; besides, diverse categories may be described by distinct features. That suggests to apply specific models on individual categories in order to reconstruct as many reply-relations as possible. Therefore, we see the reconstruction as an iterative process. When the pipeline is executed the first time, the user can observe the certainty of computed reply-relations in the previously mentioned *parent-child space*. Relations which the user agrees to be reliably reconstructed can be selected and stored in a "reliably reconstructed structure" dataset.

This stored dataset is used as a fixed structure, which is not changed by the successive executions of the pipeline (e.g., by the following usage of a classifier or a query). The user can use another query to reconstruct the remaining reply-relations. This query does not influence the reliably found and stored relations. It improves the chance to reconstruct relations where features overlap, but one of them is more descriptive than the other.

7.4 STORAGE OF THE COMPUTED STRUCTURE

One use case of the thread structure's reconstruction task is to compute a reply-relation structure from datasets where no ground-truth structure exists; the computed structure can be used for further analysis.

If the user is satisfied with the computed reply-relation structure, he can save it in a JSON file for further usage and analysis. The button for the storage is placed in the settings sidebar. For more challenging datasets, where not the whole structure can be reliably reconstructed, the user has the possibility to save only a subset of the reconstructed relations. The previously described stored relations are also written in a file. Thus, such relation subsets can be used for other use cases, like generation or expansion of specific corpus (e.g., question-answer pairs).

7.5 LIMITATIONS

The execution of the pipeline is currently done in a sequential manner. First, the classifier can be applied, then the query, and afterwards the heuristic. Such sequence is appropriate for currently trained classifiers. The precision of Decision Tree and Random Forest models is poor; thus, they can be used as a preprocessing step for the query based model. But, having a better performing classifier, the execution of single models should be more flexible. The user should be able to specify, if the classifier has to be executed before or after the query. In each case, the first applied model could be used as a preprocessing step for the subsequent model.

The stored reply-relations should be seen as a *feedback loop*. The system should learn the features present in the stored relations. Thus, the feature weights could be updated

regarding to the learned information and provided as a suggestion in the following sessions.

If the user observes one parent message being assigned wrong, he should have the chance to remove this relation or assign a new one manually. Such direct manipulation of the computed structure should also be integrated in the previously mentioned *feedback loop*.

8 EVALUATION

This chapter summarizes the evaluation results presented in Chapter 4 and Chapter 5. Three baselines are evaluated and used for a comparison. Besides, multiple use cases show the usability of the tool.

8.1 BACKGROUND OF THE TEST DATA

Like mentioned in Chapter 4 and Chapter 5, a representative sample of 40 Reddit discussions are used to evaluate the model performances. It contains 20 threads of topic "Politics", and 20 threads of topic "World News".

Fig. 24 shows the message length distribution within observed 40 threads. Three different groups are explored: messages with less than 10 tokens, messages having between 10 and 40 tokens, and messages having at least 40 tokens. In average, 51% of all messages are of a length between 10 and 40 tokens, which we categorize as "normal" messages. Both remaining groups - relatively short and relatively long messages - in average are equally distributed (almost 25% each).

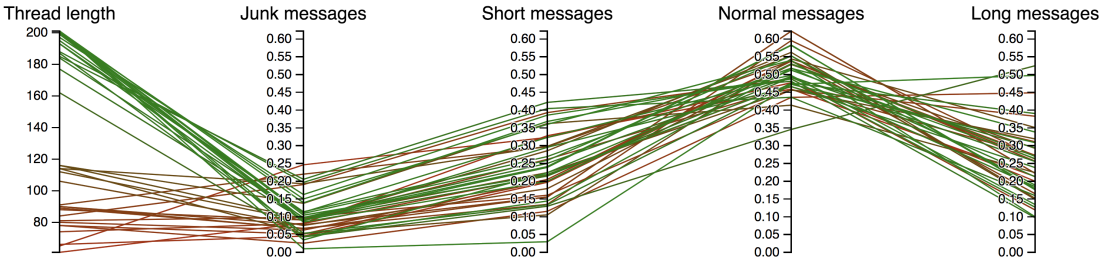


Figure 24: Parallel coordinates show an overview of the distribution of *junk* messages, and the message length in the tested 40 Reddit files.

As explained in Chapter 2, it is important to extract *junk* messages, as they lack of content information. By excluding them from dataset, the performance of the model can be improved. Like it is shown in Fig. 24, *junk* message distribution correlates with the distribution of short messages, as all messages having less than 3 tokens are classified as *junk*. 35% of all observed threads have more than 10% of messages categorized as *junk*. 10% of threads have at least 20% of *junk* messages. The reconstruction of the reply-relation structure of these threads can be challenging.

8.2 PERFORMANCE OF BASELINES

Three baseline algorithms are evaluated and used for a comparison.

REPLY TO PREVIOUS MESSAGE Balali et al. [4] show that linking a message to its previous message leads to good results for chat and forum environments. In Reddit data, in average only 6% of all messages reply to the previous message (shown in Table 14). Although this baseline may be valid for other conversational text data, it is not optimal to reconstruct reply-relations of Reddit discussions.

Files	Precision	Recall	F-Score
"Politics (60-120 msgs)"	0.02	0.02	0.02
"Politics (160-210 msgs)"	0.06	0.06	0.06
"World News (60-120 msgs)"	0.04	0.04	0.04
"World News (160-210 msgs)"	0.11	0.11	0.11
Avg.	**0.06**	**0.06**	**0.06**

Table 14: Evaluation results of the baseline: reply to the previous message.

REPLY TO TITLE MESSAGE In blogs and online news agencies the title message is considered as the parent of many following posts. In datasets such as "Thestandard", "Alef", "Narenji", used by [4], 30% of all messages reply to the title. We evaluate this baseline, and use an additional heuristic, which says that the title message can be a parent only for the particular author's first message. This baseline reaches an F-score of 0.24 (shown in Table 15).

Files	Precision	Recall	F-Score
"Politics (60-120 msgs)"	0.33	0.25	0.28
"Politics (160-210 msgs)"	0.29	0.21	0.24
"World News (60-120 msgs)"	0.22	0.18	0.20
"World News (160-210 msgs)"	0.32	0.20	0.24
Avg.	**0.30**	**0.21**	**0.24**

Table 15: Evaluation results of the baseline: reply to the title message.

CLASSIFIER + AT MOST ONE PARENT In Chapter 4 we show that the created classifiers (e.g., Random Forest, Decision Tree) alone have a relatively high recall for the "positive" class, however very low precision. When one of these classifiers is used in our tool, the system extracts at most one parent for each child message, having the highest classification score (for relations having equal scores, the first parent in the temporal order is selected). Table 16 presents that only 18% of all reply-relations can be correctly reconstructed, using the Random Forest model trained on 13 features.

Files	Precision	Recall	F-Score
"Politics (60-120 msgs)"	0.19	0.19	0.19
"Politics (160-210 msgs)"	0.20	0.20	0.20
"World News (60-120 msgs)"	0.15	0.15	0.15
"World News (160-210 msgs)"	0.18	0.19	0.18
Avg.	**0.18**	**0.18**	**0.18**

Table 16: Evaluation results of the baseline: classifier + at most one parent.

8.3 DISCUSSION: COMPARISON OF BASELINES, TRAINED CLASSIFIERS AND QUERY BASED MODEL

The evaluation of the baseline methods presents that the reconstruction of the reply-relation structure is not a trivial task. A summary of the evaluation results of different models on previously presented 40 Reddit discussions is shown in Table 17.

Model	Precision	Recall	F-Score
Baseline: reply to previous	0.06	0.06	0.06
Baseline: reply to title	0.30	0.21	0.24
Baseline: classifier	0.18	0.18	0.18
Decision Tree (5 features)	0.06	0.55	0.11
Decision Tree (13 features)	0.07	0.54	0.12
Random Forest (5 features)	0.05	0.56	0.08
Random Forest (13 features)	0.05	0.61	0.09
Query	**0.36**	**0.29**	**0.32**
***Query (threads with 30 msgs)**	**0.56**	**0.48**	**0.51**
***Query (threads with 10 msgs)**	**0.70**	**0.66**	**0.68**

Table 17: Summary of different model results. *Models are tested on cropped discussions.

The performance of the query based model is only modest (having an F-score of 0.32) applied on relatively long discussions. This performance is higher though than of the trained classifiers. We show that the query based model can compete with models presented in the related work, when it is applied on short discussions.

Multiple factors may be responsible for the poor performance of the trained classifiers, and for the modest performance of the query based model, such as the absence of reliable features, the discussion length, the quality of the input dataset, and the quality of the training dataset for the learned classifiers.

ABSENCE OF RELIABLE FEATURES The presence of reliable features has a high impact on model's performance. Datasets having many *quotes*, or many *references to*

author's name are much easier to reconstruct than those, where only less descriptive features, such as *named-entities*, are present. Only 6% of all reply-relations in the evaluated Reddit discussions have *quotes*, and less than 1% use a *reference to author's name*. In the related work, frequently authors create a model which fits exactly their used dataset. Thus, most of the models are too specific to be applied on other discussions.

The absence of reliable features is especially problematic for supervised machine learning models. If no descriptive features are present in the training data, then it is likely that the model will overfit. Due to this problem, the created models (shown in Table 17) have a low precision on the tested 40 Reddit discussions.

The query based model is more robust against this issue. Although it can only reconstruct those relations, where some descriptive features are present, the extraction of these particular relations is based on a query and not on erroneously learned rules.

DISCUSSION LENGTH The length of the used discussion is an important factor which influences the model's performance. In Table 17 we show, that by using the same query, the performance of the query based model is significantly better on short discussions than on long ones. For shorter discussions, the *distance* feature is more descriptive (shown in Chapter 4). Besides, the extracted relations are more certain, as in general there exist less parent candidates.

DATA QUALITY The overall model's performance depends on the quality of the used dataset. Forum data is a challenge for natural language processing, due to multiple factors. Unintentional errors, dialectal variation, conversational ellipsis, topic diversity, and creative use of language and orthography [16] make the processing of the conversational data difficult. Many online conversation messages are short. Frequently, such short texts cannot provide sufficient context information for similarity measure, the basis of many text processing methods [45].

Some existing work ([71], [52], [13]) present models to be used on e-mail conversations. E-mails frequently have a better quality than messages in forum discussions, as usually they imply more context information. Commonly, e-mails are used with a purpose to state a question or deliver an answer on a specific topic. Thus, reply-relations frequently have a common contextual information. In forum discussions, the content information depends on the forum's type and its participants. Two messages linked by a reply-relation not always have a contextual similarity.

Messages classified as *junk* or messages which mostly consist of URLs and special characters, have small amount of content information. Using our framework, the user is able to *clean* their dataset, by removing such noisy messages.

QUALITY OF TRAINING DATASET In order to create a qualitative supervised machine learning model, a sufficient work has to be invested in the generation of a good training dataset. The reconstruction of the reply-relation structure can be seen as an anomaly detection task, because only a small amount of all possible reply-relation candidates are existent in the ground-truth data.

One has to deal with this imbalanced classes problem; otherwise, the model could learn relations in advantage of the majority class. One possibility to obey this problem is by artificially balancing the training dataset. We do it, by applying an undersampling technique. Unfortunately, such artificial balancing of data can't always guarantee a good model's performance. Sometimes, a relevant information describing the split between two classes can get dismissed.

CONCLUSION Previously listed factors show that it is challenging to create a good model which is broadly applicable on different conversational data. Thus, a visual analytics tool is highly beneficial for the thread structure's reconstruction. Our tool can be used not only to reconstruct the structure in unseen conversations, but also to compare multiple models to each other in order to determine the best performing one. Hence, we provide a more general approach for the *RTS* task, than models presented in the related work.

8.4 USE CASES

The following use cases show the applicability of our framework. It supports two general tasks. Firstly, the user can compare multiple models to each other, when the used data has a ground truth structure. A visual evidence on the present features or the number of parent candidates describes the extracted structure's certainty. Thus, the user can select the most certain model to be applied on unseen conversations. Secondly, the user can use the framework to reconstruct the reply-relation structure in data without the ground truth. Using multiple interaction techniques (e.g., sorting of messages, *slice and dice*), the user is able to work on a specific discussion's subset at a time.

COMPARISON OF MULTIPLE MODELS/QUERIES The framework can be used to compare multiple models to each other. First of all, the user may observe which features influence different models most. The *parent-child space* provides an overview of feature distribution for reply-relation candidates. Fig. 25 shows an example of a Decision Tree model applied on a Reddit discussion. The model has extracted many parent candidates for a large number of child messages. The smooth distribution of paths indicates that many reply-relations for one child have equal scores. For equal scored relations, the system uses the temporal order to assign their position in the parent candidate list. By observing a single child message and its extracted parent candidates, one can see that most of the decisions are based on *time-distance* and *distance* features (even though among computed parent candidates exist relations, which have a relatively high *cosine similarity*). Apparently, the model has a poor performance.

The second use case shows the Random Forest model trained on 5 features (Fig. 26). Although the performance of this model is poor and many parent candidates are extracted, the performance is better than of the Decision Tree model. By observing parent candidates for one single child message, one can see that the Random Forest model makes decisions, first, based on content and structural features. That guarantees a better performance. Still, many relations having only *time-distance* and *distance* features are extracted.

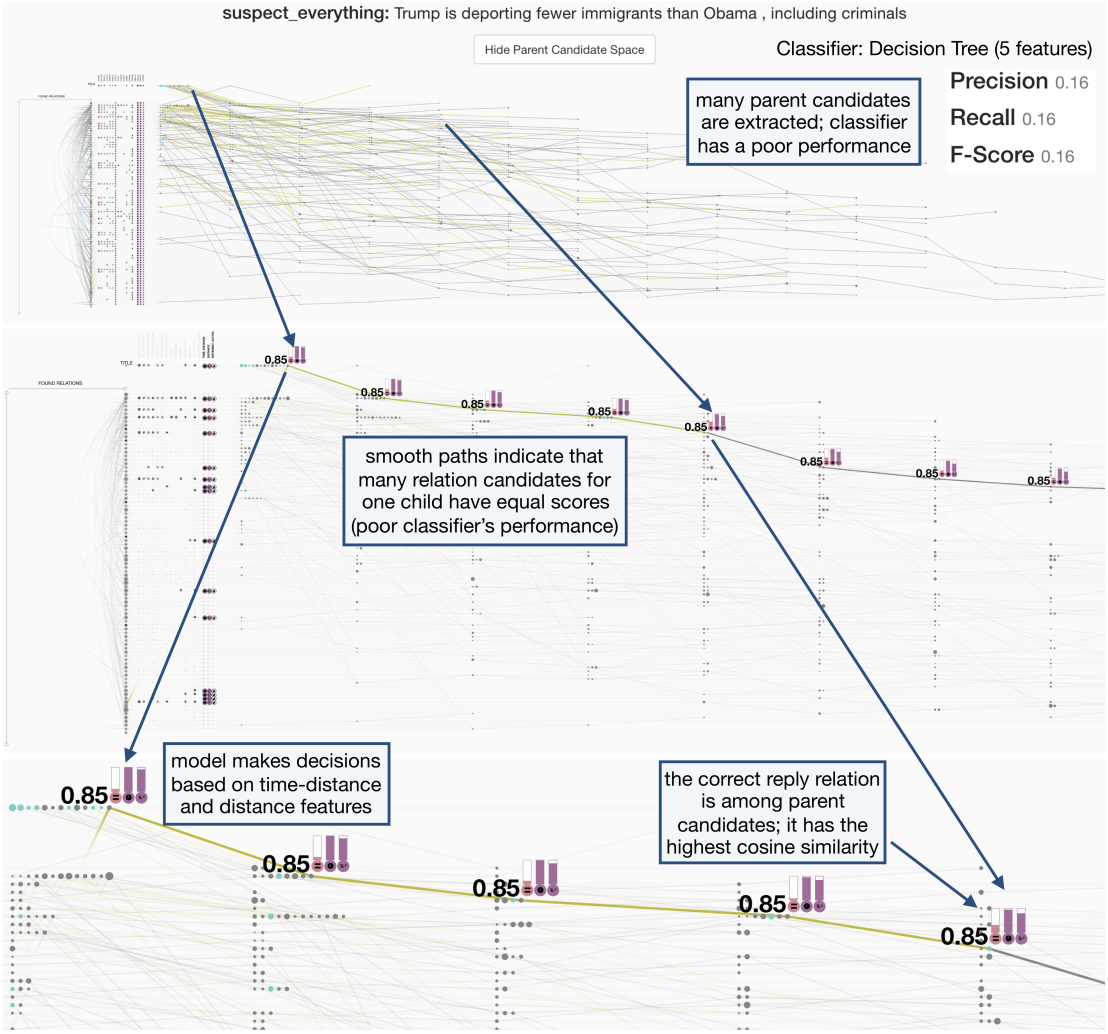


Figure 25: The 1st use case shows the low certainty of the Decision Tree model. Its decisions are based on the *distance*, and *time-distance* features.

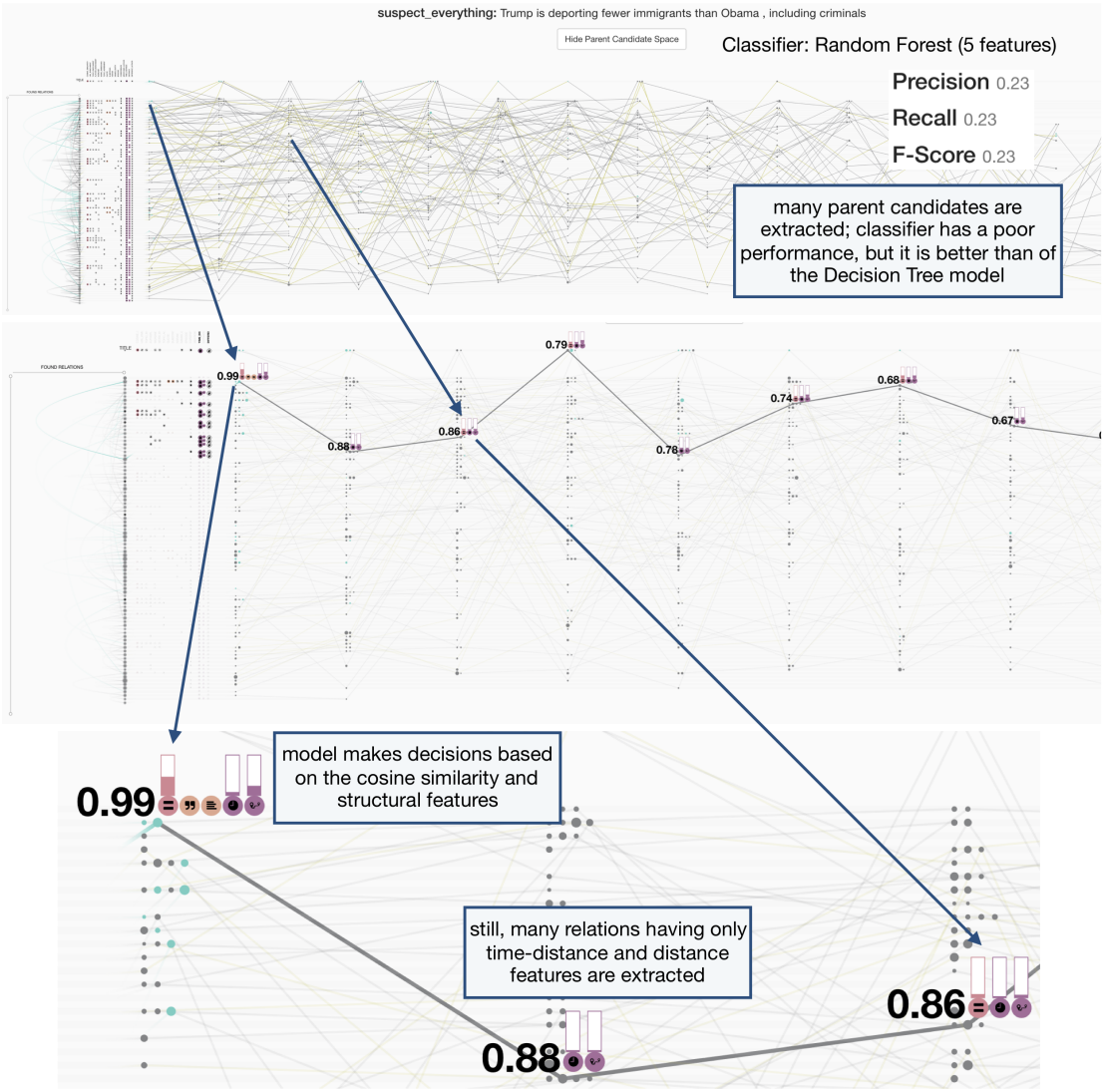


Figure 26: The 2nd use case shows a higher certainty of the Random Forest model. Its decisions are, first, based on the *cosine similarity*, and structural features. Still, many relations having only *time-distance* and *distance* features are extracted.

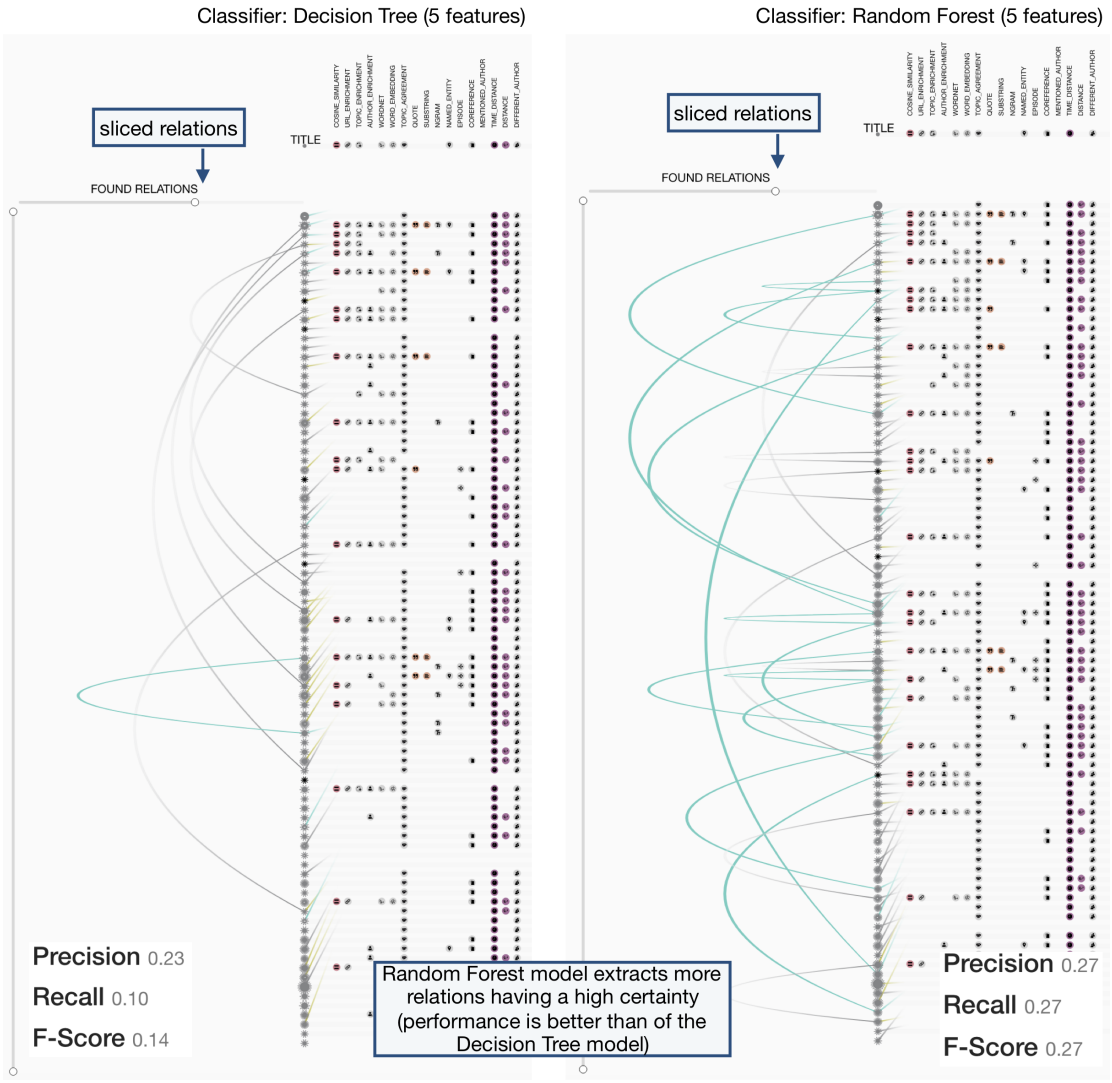


Figure 27: The 3rd use case shows a comparison between two models. Reply-relations extracted by the Random Forest model are more certain than those of the Decision Tree.

The third use case (shown in Fig. 27) presents a comparison between two different models (Decision Tree and Random Forest). The Random Forest model is able to extract more reply-relations having higher number of features. Thus, the extracted relations are more certain. The evaluation metrics strengthens this observation; the Random Forest model reaches 13% higher F-score than the Decision Tree model.

Our framework can be used not only to compare different models to each other, but also to compare different queries. Like Fig. 28 shows, not all features can reconstruct a reliable reply-relation structure. Multiple factors characterize a reliable structure. Firstly, if a computed reply-relation contains multiple features, this relation can be seen as certain. Secondly, if a relation has only meta-data or less descriptive structural features present (e.g., *coreference*), then the reliability of this relation is doubtable. Thirdly, if many child messages have multiple parent candidates, then it is more prob-

able that the system has selected the wrong parent as the most suitable one. Apparently, a query containing *cosine similarity* || *quote* || *substring* can generate a more reliable structure than a query containing *n-gram* || *named-entity*.

RECONSTRUCTION OF REPLY-RELATION STRUCTURE The framework can be used also to reconstruct the reply-relation structure of unseen conversations. To simulate the situation where a dataset without the ground truth structure is analyzed, the colors (**green** and **yellow**), which normally are used for the highlighting of correctly computed relations, are not displayed.

The fifth use case (shown in Fig. 29) presents how the user can apply different functions to improve model's performance. Like mentioned in Chapter 3, not all messages in a discussion have the same quality. The user can decide to remove the shortest messages, due to their lack of content information. When a query is executed (e.g., *cosine similarity*), the user can sort messages regarding their size (in descending order), and using the *slide and dice* technique remove the shortest messages. When messages are removed, the *parent-child space* is updated, and many parent candidates can get dismissed. Thus, the reply-relation structure gets more certain. The user can apply the *brushing* technique to select messages and store them for further usage and analysis.

The sixth use case shows an example of the iterative reconstruction's process (Fig. 30). Iterative reconstruction means that the intermediate reply-relation structure can be stored and set fixed, allowing to reconstruct relations of different message categories separately. In this example, the user executes a query (*cosine similarity* || *quote* || *substring*) and applies sorting function to order messages regarding their word count. Then, using the *slice and dice* technique, short messages are removed from the *parent child space*. Apparently, the remaining reply-relations are certain, having many content features present. These relations can be stored and set fixed. Then, another query (*n-gram* || *named-entity*) can be applied on the rest of the discussion. Stored relations are not influenced by this query. The F-score of the model applied on long messages can be improved.

The last use case shows an example of a short discussion's summary displayed in the *disentangled forest view* (shown in Fig. 31). There, found connected components are sorted to their size and moved together. This view gives an insight into the discussed subtopics.

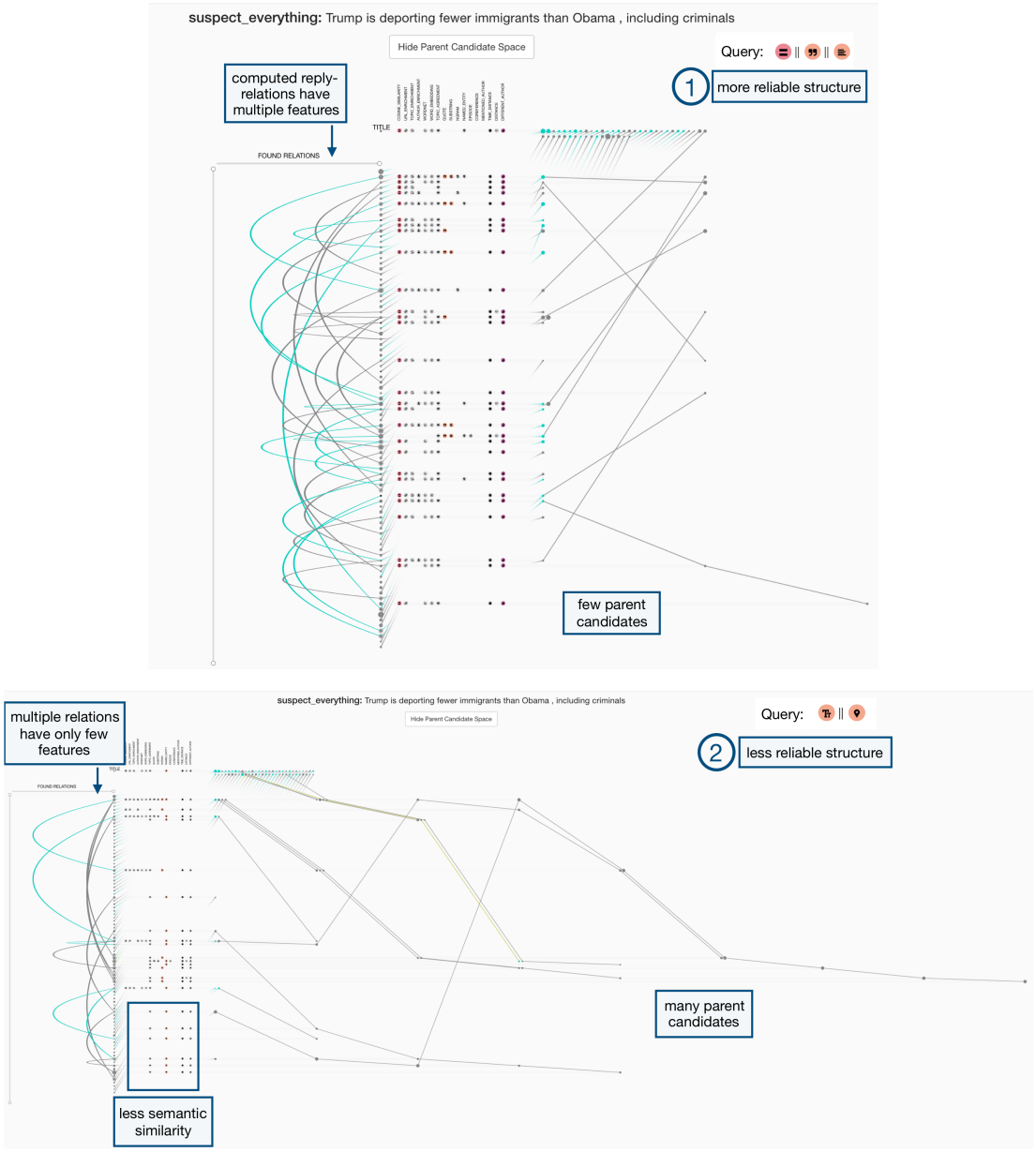


Figure 28: The 4th use case shows a comparison between the results of two queries. A query consisting of *cosine similarity*, *quote*, and *substring* features can generate a more reliable structure than a query of *n-gram* and *named-entity*.

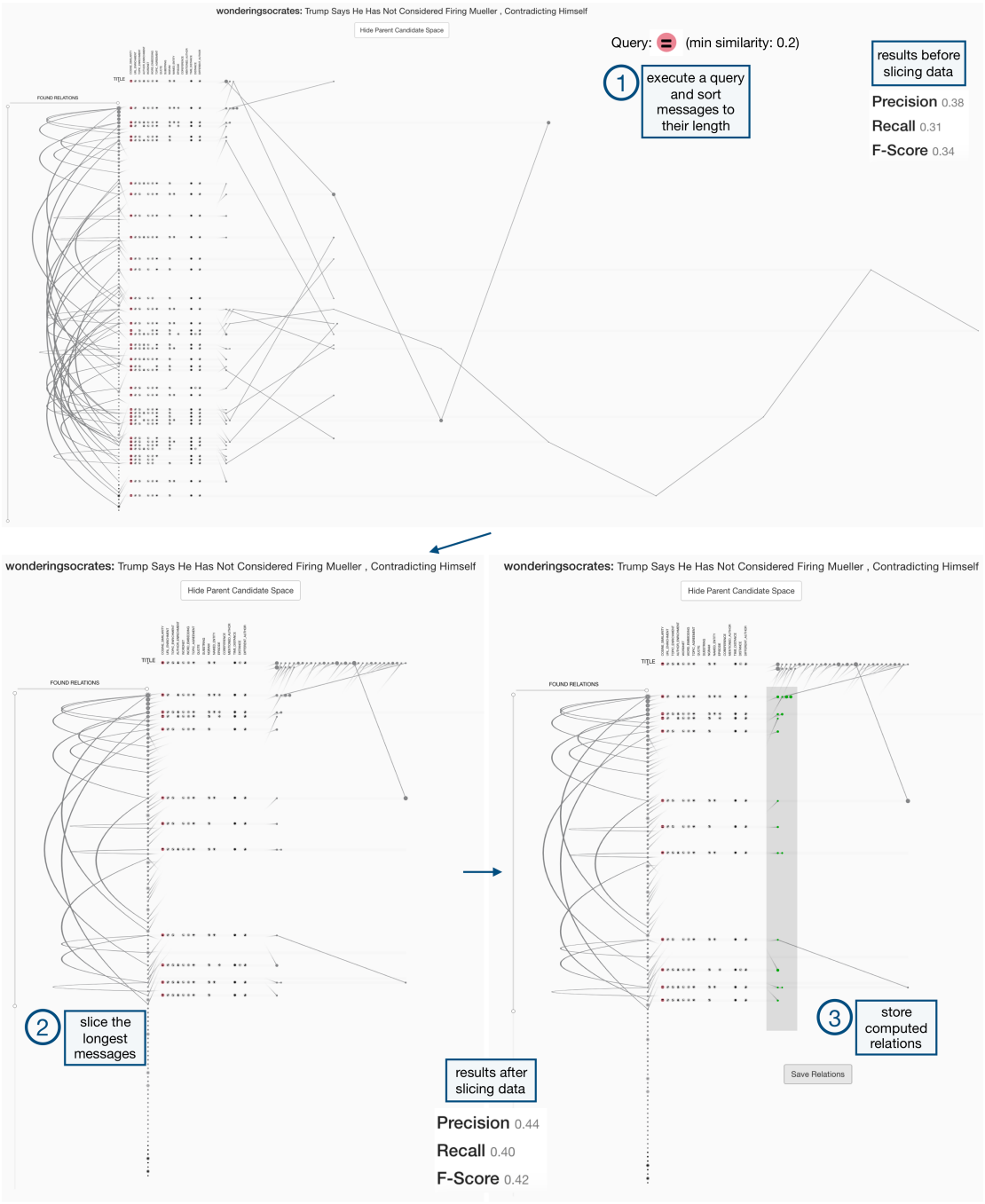


Figure 29: The 5th use case shows how multiple functions can be applied to improve the model's performance.

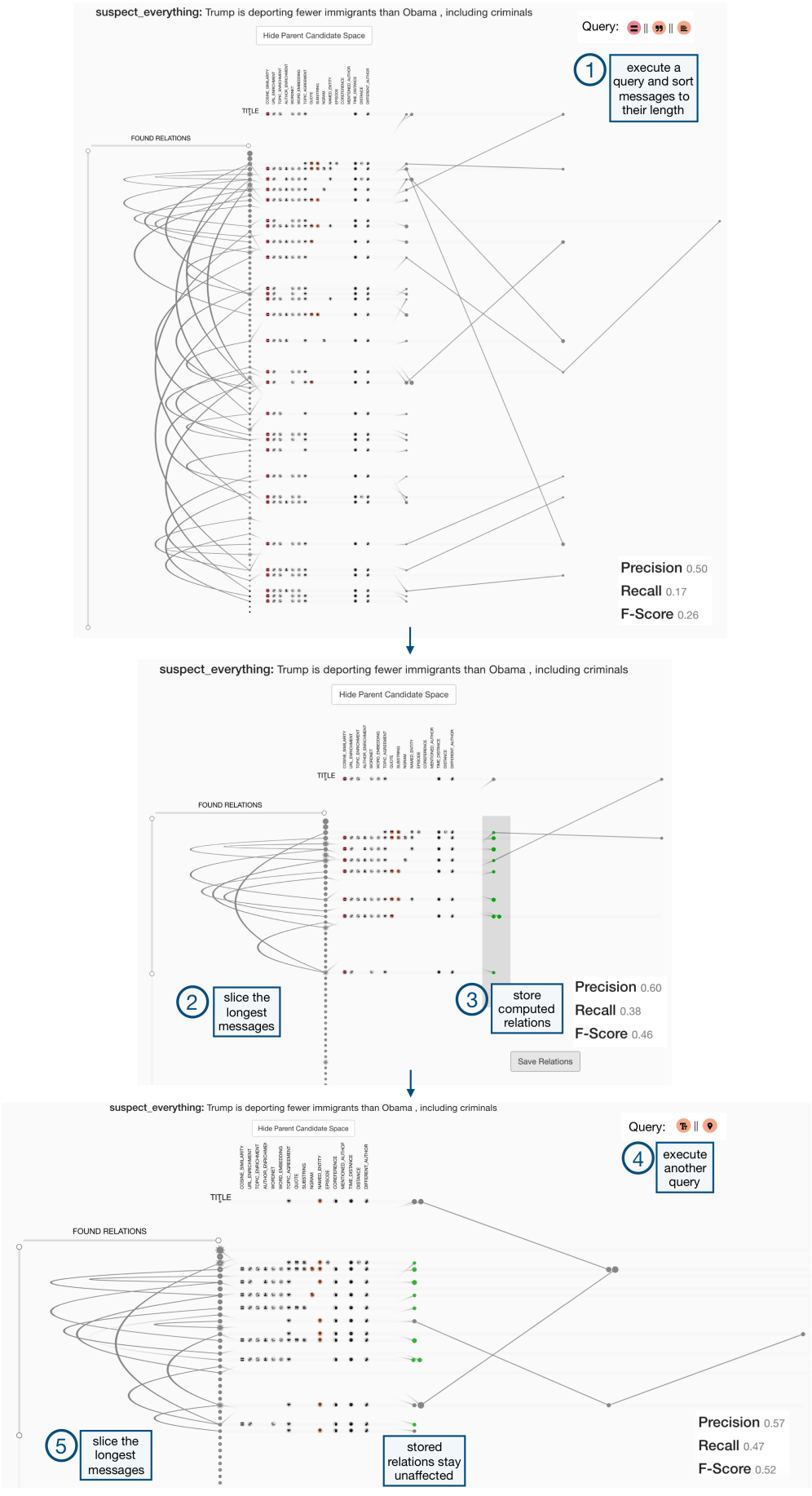


Figure 30: The 6th use case shows an example of the iterative reconstruction's process. Different queries can be applied on different message categories.

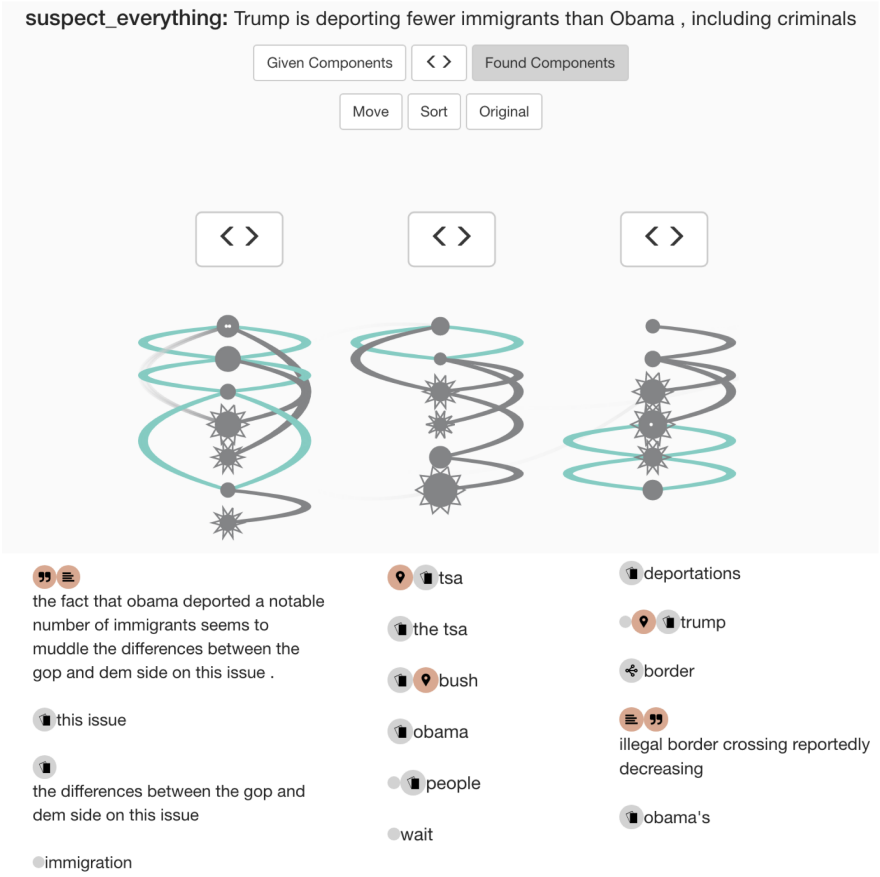


Figure 31: The 7th use case shows an example of a short discussion's summary in the *disentangled forest view*.

9 CONCLUSIONS

This thesis presents a visual analytics approach to reconstruct the reply-relation structure of conversational text data. Models presented in the prior research suffer from multiple drawbacks. Most of the models have a restricted applicability; they are trained on a specific dataset. Thus, these models are applicable only on datasets which have similar characteristics. All existing models are evaluated using statistical metrics, like precision and recall. Sometimes, such evaluation metrics alone may be misleading, especially if the supervised model has overfitted the training data.

To address the previously mentioned issues, we provide a visual analytics tool, which combines a machine learning model with an unsupervised query based model, and is more general and flexible approach than the existing methods. We train multiple supervised models (e.g., Decision Tree, Random Forest) using different sets of features. The supervised models are evaluated in two steps. First, a k-fold cross-validation technique is applied to evaluate their performance. The performance is satisfying; the Decision Tree model trained on 13 features reaches an F-score of 0.73. In the second step, models are tested on a representative sample of 40 Reddit discussions, which are not used in the training process. The performance of the models decreases significantly (the average precision reaches only 0.06). Apparently, they have overfitted the training data. The query based model outperforms the trained classifiers, despite its modest F-score of 0.32. We show, that its performance increases, when the model is applied on short discussions. The query based model is flexible; 17 different features can be combined to generate a representative query.

The poor performance of the trained classifiers may have multiple reasons. The reply-relation reconstruction of relatively long discussions is similar to an anomaly detection task; only small part of all possible reply-relation candidates are given in the ground truth structure. Usually, a machine learning algorithm requires a balanced training data in order to generate a well performing model. When data is artificially balanced using under-sampling technique, many instances representing the majority class are dismissed. Hence, some relevant information describing the split between two classes may get lost. Besides, due to a lack of reliable features in the used dataset, supervised models tend to overfit. The unsupervised query based model is more robust against these issues.

Our framework uses the user's knowledge and automatic data analysis methods to reconstruct the reply-relation structure of an input discussion, and provides suggestions for the most descriptive features and their default parameters. It produces an evidence on the computed structure's certainty. The *parent-child space* visualization can be used to compare multiple models to each other, or to observe the certainty level of the extracted structure having no ground truth information. The tool supports *close reading*, and disentangles the discussion's main subtopics.

Multiple use cases, presented in Chapter 8, show the applicability of our framework. Nevertheless, the extension of the framework is desirable. The execution of the pipeline should be configured to allow the user to change the order of individual steps (e.g., to execute the query before the classifier). In order to improve the detection of patterns, when different classifiers are compared to each other, the visual representation should provide an additional evidence on the feature influence on each instance separately. A better representation of content feature values for each candidate relation is needed. More suggestions for improvements are listed in Chapter 10.

10 FUTURE WORK

The execution of the pipeline is currently done in a sequential manner. First, the classifier can be applied, then the query, and afterwards the heuristic. For better performing classifiers, the compilation of the pipeline should be more flexible. The user should be able to specify, if the classifier has to be executed before or after the query. In each case, the first applied model could be used as a preprocessing step for the subsequent model.

The system should provide more suggestions, which features are descriptive for the used dataset. Currently, the frequency of the present features in all candidate relations is shown as a tooltip. Instead of the frequency, an entropy could be used to present the feature descriptiveness.

The stored reply-relations should be seen as a *feedback loop*. The system should learn the features present in the stored relations. Thus, the feature weights could be updated regarding the learned information and provided as a suggestion in the following sessions.

If the user observes one parent message being assigned wrong, he should be able to remove this relation and assign a new one manually, if needed.

Additional features, like extraction of question-answer pairs might improve the performance of the query based model. Like it is shown in Chapter 3, 8% of all observed messages in 10 Reddit discussions end with a question mark. Thus, a technique to extract the question-answer pairs is highly beneficial.

A better representation of the content feature scores for each candidate relation is needed. Currently, an overview of present features in the reply-relation candidates are highlighted in the *parent-child space*. A more detailed representation showing exact values and their changes would improve the readability of the visualization.

Besides, the representation of the model's reliability should be improved. Currently, the **green** and **yellow** colors are used to highlight child-paths, where the correct parent is among the parent candidates. It might be useful to visually represent if the transitive reply-relations are given in the ground truth structure. If the system detects a message as a parent candidate, but in the ground truth structure the child or the parent of the detected message is the searched parent, it may indicate, that the model's reconstructed structure is relatively close to the ground truth.

The *disentangled forest view* can be used to compare the given and the computed connected components to each other (if the used data has a ground truth structure), or to observe the computed discussion's subtopics in the dataset without the ground truth information. Regarding the first use case, two *disentangled forests* (given and

computed) could be displayed simultaneously, highlighting components which have some reply-relation matches. Such representation could give an additional insight, which subtopics are easier to be reconstructed.

The representation of the common phrases for one connected component (in the *disentangled forest view*) should be improved. The exact position of the word or phrase should be highlighted in the disentangled discussion's structure. The challenge is to avoid a clutter of the represented data.

To analyse a new Reddit discussion, the system should allow the user to input a thread's URL; the wrapping of the discussion's structure should be done automatically. It would improve the usability of the tool.

A

APPENDIX

A.1 EVALUATION RESULTS

The following tables present the evaluation results of different models, tested on 40 Reddit discussions. Listed are results for each discussion separately. Table 18 shows results of the Random Forest model trained on an imbalanced training dataset. Table 19 presents results of the Decision Tree model trained on a balanced training dataset using 5 features presented by [3]. Table 20 shows evaluation results of the Decision Tree model trained on a balanced training set using 13 features. Table 21 shows evaluation results of the query based model (using query: ((*quote* (weight: 5) || *substring* (weight: 4) || *cosine similarity* (min similarity: 0.2, weight: 3) && *different author*) || *time-distance* (max-distance: 24 hours, weight:1)). Tables 22, 23, and 24 list results of three baselines, presented in 8.

File	Precision (p)	Recall (p)	F-score (p)	Precision (n)	Recall (n)	F-score (n)
Politics 1	1	0.01	0.02	0.99	1	1
Politics 2	1	0.01	0.01	0.99	1	1
Politics 3	0.67	0.01	0.02	0.99	1	1
Politics 4	0	0	0	0.99	1	0.99
Politics 5	0.5	0.01	0.01	0.99	1	0.99
Politics 6	0	0	0	0.99	1	0.99
Politics 7	0.67	0.01	0.02	0.99	1	0.99
Politics 8	1	0.01	0.02	0.99	1	0.99
Politics 9	0	0	0	0.99	1	0.99
Politics 10	0.75	0.02	0.03	0.99	1	1
Politics 11	0	0	0	0.98	1	0.99
Politics 12	0	0	0	0.98	1	0.99
Politics 13	0	0	0	0.98	1	0.99
Politics 14	1	0.01	0.02	0.98	1	0.99
Politics 15	0	0	0	0.97	1	0.99
Politics 16	0.75	0.03	0.05	0.98	1	0.99
Politics 17	0	0	0	0.98	1	0.99
Politics 18	0	0	0	0.98	1	0.99
Politics 19	1	0.01	0.02	0.98	1	0.99
Politics 20	0	0	0	0.98	1	0.99
WolrdNews 1	0.5	0.01	0.01	0.99	1	0.99
WorldNews 2	1	0.01	0.01	0.99	1	1
WorldNews 3	1	0.01	0.01	0.99	1	0.99
WorldNews 4	0.5	0.01	0.02	0.99	1	0.99
WorldNews 5	1	0.01	0.02	0.99	1	0.99
WorldNews 6	1	0.02	0.03	0.99	1	0.99
WorldNews 7	1	0.01	0.01	0.99	1	0.99
WorldNews 8	1	0.01	0.03	0.99	1	0.99
WorldNews 9	0	0	0	0.99	1	0.99
WorldNews 10	0.5	0.01	0.01	0.99	1	0.99
WorldNews 11	1	0.04	0.08	0.98	1	0.99
WorldNews 12	0	0	0	0.98	1	0.99
WorldNews 13	1	0.02	0.03	0.97	1	0.99
WorldNews 14	1	0.01	0.03	0.98	1	0.99
WorldNews 15	0	0	0	0.97	1	0.98
WorldNews 16	0	0	0	0.98	1	0.99
WorldNews 17	1	0.01	0.02	0.98	1	0.99
WorldNews 18	1	0.01	0.03	0.97	1	0.99
WorldNews 19	0	0	0	0.97	1	0.99
WorldNews 20	1	0.01	0.02	0.98	1	0.99
Avg.	**0.55**	**0.01**	**0.01**	**0.98**	**1**	**0.99**

Table 18: Evaluation results of the Random Forest model trained on an imbalanced training dataset, and tested on 40 test discussions. (*p-"positive" class, n-"negative" class)

File	Precision (p)	Recall (p)	F-score (p)	Precision (n)	Recall(n)	F-score(n)
Politics 1	0.03	0.48	0.05	0.99	0.83	0.91
Politics 2	0.03	0.5	0.06	0.99	0.85	0.92
Politics 3	0.04	0.5	0.08	0.99	0.89	0.94
Politics 4	0.04	0.56	0.08	0.99	0.88	0.93
Politics 5	0.02	0.43	0.04	0.99	0.82	0.9
Politics 6	0.03	0.54	0.06	0.99	0.85	0.92
Politics 7	0.04	0.5	0.07	0.99	0.87	0.93
Politics 8	0.04	0.49	0.07	0.99	0.88	0.93
Politics 9	0.03	0.45	0.06	0.99	0.87	0.93
Politics 10	0.04	0.54	0.08	0.99	0.87	0.93
Politics 11	0.06	0.46	0.11	0.99	0.84	0.91
Politics 12	0.07	0.63	0.13	0.99	0.81	0.89
Politics 13	0.08	0.57	0.14	0.99	0.88	0.93
Politics 14	0.07	0.52	0.12	0.99	0.84	0.91
Politics 15	0.1	0.6	0.17	0.99	0.86	0.92
Politics 16	0.07	0.65	0.13	0.99	0.85	0.92
Politics 17	0.1	0.59	0.17	0.99	0.86	0.92
Politics 18	0.08	0.56	0.14	0.99	0.89	0.94
Politics 19	0.07	0.52	0.12	0.99	0.84	0.91
Politics 20	0.05	0.58	0.09	0.99	0.79	0.88
WolrdNews 1	0.05	0.55	0.08	0.99	0.88	0.94
WorldNews 2	0.04	0.45	0.07	0.99	0.89	0.94
WorldNews 3	0.05	0.53	0.09	0.99	0.89	0.94
WorldNews 4	0.04	0.51	0.08	0.99	0.87	0.93
WorldNews 5	0.05	0.48	0.08	0.99	0.89	0.94
WorldNews 6	0.04	0.55	0.08	0.99	0.86	0.93
WorldNews 7	0.04	0.61	0.07	1	0.84	0.91
WorldNews 8	0.04	0.58	0.08	0.99	0.85	0.92
WorldNews 9	0.04	0.48	0.08	0.99	0.88	0.94
WorldNews 10	0.04	0.52	0.07	0.99	0.86	0.92
WorldNews 11	0.1	0.74	0.18	0.99	0.83	0.9
WorldNews 12	0.06	0.47	0.11	0.99	0.87	0.92
WorldNews 13	0.1	0.53	0.16	0.98	0.85	0.91
WorldNews 14	0.08	0.58	0.14	0.99	0.83	0.9
WorldNews 15	0.11	0.58	0.18	0.98	0.84	0.9
WorldNews 16	0.08	0.55	0.14	0.99	0.85	0.91
WorldNews 17	0.09	0.61	0.16	0.99	0.86	0.92
WorldNews 18	0.09	0.62	0.15	0.99	0.82	0.9
WorldNews 19	0.1	0.75	0.18	0.99	0.81	0.89
WorldNews 20	0.07	0.61	0.13	0.99	0.86	0.92
Avg.	**0.06**	**0.55**	**0.11**	**0.99**	**0.86**	**0.92**

Table 19: Evaluation results of the Decision Tree model trained on a balanced training dataset using 5 features, and tested on 40 test discussions. (*p-"positive" class, n-"negative" class)

File	Precision (p)	Recall (p)	F-score (p)	Precision (n)	Recall(n)	F-score(n)
Politics 1	0.03	0.52	0.06	0.99	0.85	0.92
Politics 2	0.03	0.51	0.06	0.99	0.84	0.91
Politics 3	0.04	0.51	0.08	0.99	0.89	0.94
Politics 4	0.05	0.56	0.08	1	0.88	0.93
Politics 5	0.02	0.45	0.04	0.99	0.79	0.88
Politics 6	0.03	0.55	0.06	0.99	0.84	0.91
Politics 7	0.04	0.52	0.08	0.99	0.87	0.93
Politics 8	0.04	0.51	0.08	0.99	0.88	0.93
Politics 9	0.04	0.48	0.07	0.99	0.87	0.93
Politics 10	0.04	0.55	0.08	0.99	0.87	0.93
Politics 11	0.09	0.42	0.14	0.99	0.9	0.94
Politics 12	0.09	0.61	0.15	0.99	0.85	0.91
Politics 13	0.09	0.52	0.16	0.99	0.91	0.95
Politics 14	0.08	0.5	0.14	0.99	0.87	0.93
Politics 15	0.12	0.57	0.2	0.99	0.89	0.94
Politics 16	0.06	0.63	0.11	0.99	0.83	0.91
Politics 17	0.11	0.54	0.19	0.99	0.89	0.94
Politics 18	0.09	0.52	0.16	0.99	0.91	0.95
Politics 19	0.08	0.5	0.14	0.99	0.87	0.93
Politics 20	0.06	0.58	0.1	0.99	0.82	0.9
WolrdNews 1	0.05	0.57	0.09	1	0.88	0.94
WorldNews 2	0.04	0.45	0.07	0.99	0.89	0.94
WorldNews 3	0.05	0.54	0.09	0.99	0.89	0.94
WorldNews 4	0.04	0.52	0.08	0.99	0.88	0.93
WorldNews 5	0.05	0.47	0.09	0.99	0.9	0.94
WorldNews 6	0.05	0.55	0.09	0.99	0.88	0.93
WorldNews 7	0.04	0.62	0.08	1	0.85	0.91
WorldNews 8	0.04	0.6	0.08	0.99	0.85	0.91
WorldNews 9	0.04	0.47	0.08	0.99	0.89	0.94
WorldNews 10	0.04	0.52	0.07	0.99	0.86	0.92
WorldNews 11	0.12	0.66	0.2	0.99	0.87	0.93
WorldNews 12	0.07	0.43	0.12	0.99	0.89	0.94
WorldNews 13	0.12	0.48	0.2	0.98	0.89	0.94
WorldNews 14	0.09	0.59	0.16	0.99	0.85	0.91
WorldNews 15	0.12	0.65	0.2	0.99	0.84	0.9
WorldNews 16	0.1	0.49	0.16	0.99	0.89	0.94
WorldNews 17	0.1	0.58	0.17	0.99	0.88	0.93
WorldNews 18	0.11	0.68	0.19	0.99	0.85	0.91
WorldNews 19	0.11	0.73	0.19	0.99	0.82	0.9
WorldNews 20	0.07	0.63	0.12	0.99	0.84	0.91
Avg.	**0.07**	**0.54**	**0.12**	**0.99**	**0.87**	**0.93**

Table 20: Evaluation results of the Decision Tree model trained on a balanced training dataset using 13 features, and tested on 40 test discussions. (*p-"positive" class, n-"negative" class)

File	Precision	Recall	F-score
Politics 1	0.73	0.5	0.59
Politics 2	0.36	0.32	0.34
Politics 3	0.28	0.24	0.26
Politics 4	0.28	0.25	0.27
Politics 5	0.37	0.3	0.33
Politics 6	0.28	0.25	0.26
Politics 7	0.3	0.26	0.28
Politics 8	0.28	0.25	0.26
Politics 9	0.32	0.29	0.3
Politics 10	0.38	0.33	0.35
Politics 11	0.26	0.22	0.24
Politics 12	0.63	0.43	0.51
Politics 13	0.41	0.31	0.35
Politics 14	0.43	0.37	0.4
Politics 15	0.3	0.23	0.26
Politics 16	0.36	0.3	0.33
Politics 17	0.29	0.23	0.26
Politics 18	0.4	0.31	0.35
Politics 19	0.43	0.37	0.4
Politics 20	0.35	0.21	0.27
WolrdNews 1	0.23	0.22	0.22
WorldNews 2	0.23	0.22	0.23
WorldNews 3	0.21	0.21	0.21
WorldNews 4	0.32	0.27	0.29
WorldNews 5	0.27	0.22	0.24
WorldNews 6	0.44	0.34	0.38
WorldNews 7	0.33	0.26	0.29
WorldNews 8	0.31	0.31	0.31
WorldNews 9	0.24	0.22	0.23
WorldNews 10	0.28	0.24	0.26
WorldNews 11	0.4	0.29	0.33
WorldNews 12	0.46	0.34	0.39
WorldNews 13	0.36	0.32	0.34
WorldNews 15	0.24	0.18	0.21
WorldNews 14	0.42	0.33	0.37
WorldNews 16	0.52	0.44	0.48
WorldNews 17	0.42	0.26	0.32
WorldNews 18	0.43	0.22	0.29
WorldNews 19	0.4	0.3	0.34
WorldNews 20	0.35	0.28	0.31
Avg.	**0.36**	**0.29**	**0.32**

Table 21: Evaluation results of the query based model (using query ((*quote* (weight: 5) || *substring* (weight: 4) || *cosine similarity* (min similarity: 0.2, weight: 3) && *different author*)||*time-distance* (max-distance: 24 hours, weight: 1)).

File	Precision	Recall	F-score
Politics 1	0.76	0.43	0.55
Politics 2	0.29	0.24	0.26
Politics 3	0.21	0.16	0.18
Politics 4	0.23	0.19	0.21
Politics 5	0.37	0.28	0.32
Politics 6	0.22	0.19	0.20
Politics 7	0.36	0.28	0.31
Politics 8	0.20	0.15	0.17
Politics 9	0.36	0.30	0.33
Politics 10	0.30	0.24	0.27
Politics 11	0.16	0.12	0.14
Politics 12	0.52	0.30	0.38
Politics 13	0.30	0.21	0.24
Politics 14	0.34	0.28	0.31
Politics 15	0.23	0.17	0.19
Politics 16	0.31	0.23	0.26
Politics 17	0.22	0.16	0.19
Politics 18	0.29	0.20	0.24
Politics 19	0.34	0.28	0.31
Politics 20	0.23	0.10	0.14
WolrdNews 1	0.13	0.13	0.130
WorldNews 2	0.19	0.17	0.18
WorldNews 3	0.09	0.08	0.09
WorldNews 4	0.32	0.24	0.27
WorldNews 5	0.17	0.14	0.15
WorldNews 6	0.37	0.23	0.28
WorldNews 7	0.23	0.14	0.18
WorldNews 8	0.24	0.24	0.24
WorldNews 9	0.17	0.15	0.16
WorldNews 10	0.30	0.24	0.27
WorldNews 11	0.21	0.12	0.15
WorldNews 12	0.42	0.27	0.33
WorldNews 13	0.39	0.34	0.36
WorldNews 14	0.27	0.16	0.20
WorldNews 15	0.28	0.18	0.22
WorldNews 16	0.51	0.39	0.44
WorldNews 17	0.24	0.13	0.17
WorldNews 18	0.41	0.12	0.19
WorldNews 19	0.26	0.12	0.16
WorldNews 20	0.21	0.14	0.17
Avg.	**0.295**	**0.209**	**0.242**

Table 22: Evaluation results of the baseline: reply to the title message.

File	Precision	Recall	F-score
Politics 1	0.01	0.01	0.01
Politics 2	0.01	0.01	0.01
Politics 3	0.04	0.04	0.04
Politics 4	0.03	0.03	0.03
Politics 5	0.01	0.01	0.01
Politics 6	0.01	0.01	0.01
Politics 7	0.04	0.04	0.04
Politics 8	0.02	0.02	0.02
Politics 9	0.02	0.02	0.02
Politics 10	0.03	0.03	0.03
Politics 11	0.03	0.03	0.03
Politics 12	0.03	0.03	0.03
Politics 13	0.07	0.08	0.07
Politics 14	0.08	0.08	0.08
Politics 15	0.09	0.09	0.09
Politics 16	0.04	0.04	0.04
Politics 17	0.09	0.09	0.09
Politics 18	0.07	0.07	0.07
Politics 19	0.08	0.08	0.08
Politics 20	0.01	0.01	0.01
WolrdNews 1	0.01	0.01	0.01
WorldNews 2	0.04	0.04	0.04
WorldNews 3	0.04	0.04	0.04
WorldNews 4	0.04	0.04	0.04
WorldNews 5	0.03	0.03	0.03
WorldNews 6	0.09	0.09	0.09
WorldNews 7	0.04	0.04	0.04
WorldNews 8	0.01	0.01	0.01
WorldNews 9	0.04	0.04	0.04
WorldNews 10	0.02	0.02	0.02
WorldNews 11	0.13	0.13	0.13
WorldNews 12	0.03	0.03	0.03
WorldNews 13	0.08	0.08	0.08
WorldNews 14	0.11	0.11	0.11
WorldNews 15	0.13	0.13	0.13
WorldNews 16	0.07	0.07	0.07
WorldNews 17	0.18	0.19	0.19
WorldNews 18	0.14	0.14	0.14
WorldNews 19	0.11	0.12	0.11
WorldNews 20	0.07	0.07	0.07
Avg.	**0.06**	**0.06**	**0.06**

Table 23: Evaluation results of the baseline: reply to the previous message.

File	Precision	Recall	F-score
Politics 1	0.28	0.28	0.28
Politics 2	0.24	0.24	0.24
Politics 3	0.17	0.17	0.17
Politics 4	0.14	0.14	0.14
Politics 5	0.15	0.15	0.15
Politics 6	0.18	0.18	0.18
Politics 7	0.20	0.21	0.20
Politics 8	0.15	0.15	0.15
Politics 9	0.22	0.22	0.22
Politics 10	0.14	0.14	0.14
Politics 11	0.19	0.19	0.19
Politics 12	0.27	0.27	0.27
Politics 13	0.18	0.19	0.18
Politics 14	0.26	0.27	0.26
Politics 15	0.18	0.18	0.18
Politics 16	0.15	0.15	0.15
Politics 17	0.18	0.18	0.18
Politics 18	0.18	0.19	0.18
Politics 19	0.26	0.27	0.26
Politics 20	0.10	0.10	0.10
WolrdNews 1	0.16	0.16	0.16
WorldNews 2	0.14	0.14	0.14
WorldNews 3	0.10	0.10	0.10
WorldNews 4	0.13	0.13	0.13
WorldNews 5	0.14	0.14	0.14
WorldNews 6	0.14	0.14	0.14
WorldNews 7	0.15	0.15	0.15
WorldNews 8	0.17	0.18	0.17
WorldNews 9	0.12	0.12	0.12
WorldNews 10	0.20	0.20	0.20
WorldNews 11	0.09	0.09	0.09
WorldNews 12	0.23	0.23	0.23
WorldNews 13	0.30	0.31	0.30
WorldNews 14	0.16	0.16	0.16
WorldNews 15	0.23	0.23	0.2
WorldNews 16	0.29	0.29	0.29
WorldNews 17	0.16	0.17	0.16
WorldNews 18	0.12	0.12	0.12
WorldNews 19	0.11	0.12	0.11
WorldNews 20	0.15	0.15	0.15
Avg.	**0.182**	**0.184**	**0.182**

Table 24: Evaluation results of the baseline: classifier (each message has only one parent candidate).

BIBLIOGRAPHY

[1] C. C. Aggarwal and C. X. Zhai. Mining text data. *Mining Text Data*:1–522, 2013. ISSN: 1098-6596. DOI: 10.1007/978-1-4614-3223-4.

[2] M. Assady. *Incremental Hierarchical Topic Modeling for Multi-party Conversation Analysis*. 2015. URL: https://books.google.ch/books?id=Yw8OjwEACAAJ.

[3] E. Aumayr and J. Chan. Reconstruction of Threaded Conversations in Online Discussion Forums. *Artificial Intelligence*:26–33, 2011. URL: http://www.aaai.org/ocs/index.php/ICWSM/ICWSM11/paper/viewPDFInterstitial/2840/3279.

[4] A. Balali, H. Faili, M. Asadpour, and M. Dehghani. A supervised approach for reconstructing thread structure in comments on blogs and online news agencies. *Computacion y Sistemas*, 17(2):207–217, 2013. ISSN: 14055546.

[5] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent Dirichlet Allocation. *The Journal of Machine Learning Research*, 3(3):993–1022, 2003. ISSN: 1532-4435. DOI: http://dx.doi.org/10.1162/jmlr.2003.3.4-5.993.

[6] L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001. ISSN: 08856125. DOI: 10.1023/A:1010933404324.

[7] R. Caruana, Y. Lou, J. Gehrke, P. Koch, M. Sturm, and N. Elhadad. Intelligible Models for HealthCare. *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '15*:1721–1730, 2015. DOI: 10.1145/2783258.2788613. URL: http://dl.acm.org/citation.cfm?id=2783258.2788613.

[8] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. SMOTE: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16:321–357, 2002. ISSN: 10769757. DOI: 10.1613/jair.953.

[9] C Chemudugunta, P Smyth, and M Steyvers. Modeling general and specific aspects of documents with a probabilistic topic model. *Nips*, 19:241–248, 2006. ISSN: 10495258.

[10] C. Cortes and V. Vapnik. Support-Vector Networks. *Machine Learning*, 20(3):273–297, 1995. ISSN: 15730565. DOI: 10.1023/A:1022627411411.

[11] M. Daszykowski and B. Walczak. Density-Based Clustering Methods. *Comprehensive Chemometrics*, 2:635–654, 2010. ISSN: 09758887. DOI: 10.1016/B978-044452701-1.00067-3.

[12] K. Dave, M. Wattenberg, and M. Muller. IBM Research Report: Flash Forums and ForumReader : Navigating a New Kind of Large-Scale Online Discussion. *IBM Watson Research Center*, 23305:1–11, 2004. DOI: 10.1145/1031607.1031644.

[13] M. Dehghani, a. Shakery, M. Asadpour, and a. Koushkestani. A learning approach for email conversation thread reconstruction. *Journal of Information Science*, 39(6):846–863, 2013. ISSN: 0165-5515. DOI: 10.1177/0165551513494638. URL: http://jis.sagepub.com/cgi/doi/10.1177/0165551513494638.

[14] G. Domeniconi, K. Semertzidis, V. Lopez, E. Daly, S. Kotoulas, and G. Moro. A novel method for unsupervised and supervised conversational message thread detection. *DATA 2016 - Proceedings of the 5th International Conference on Data Management Technologies and Applications*, (July):978–989, 2016. DOI: 10.5220/0006001100430054.

[15] P. Domingos. MetaCost: A General Method for Making Classifiers Cost-Sensitive. *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, 55:155–164, 1999. ISSN: 1550-4786. DOI: 10.1145/312129.312220. URL: http://portal.acm.org/citation.cfm?id=312129.312220&type=series.

[16] J. Eisenstein. What to do about bad language on the internet. *Naacl-Hlt 2013*:359–369, 2013.

[17] A. Ekbal and S. Saha. Weighted vote-based classifier ensemble for named entity recognition: a genetic algorithm-based approach. *ACM Trans. Asian Lang. Inf. Process.*, 10:9:1–9:37, 2011.

[18] C. Elkan. The foundations of cost-sensitive learning. *IJCAI International Joint Conference on Artificial Intelligence*:973–978, 2001. ISSN: 10450823. DOI: doi=10.1.1.29.514.

[19] J. Elsas and J. Carbonell. It pays to be picky: an evaluation of thread retrieval in online forums. *Sigir'09*, (2):714–715, 2009. DOI: 10.1145/1571941.1572092. URL: http://dl.acm.org/citation.cfm?id=1572092.

[20] J. L. Elsas, J. Arguello, J. Callan, and J. G. Carbonell. Retrieval and {Feedback} {Models} for {Blog} {Feed} {Search}. *Proceedings of the 31st {Annual} {International} {ACM} {SIGIR} {Conference} on {Research} and {Development} in {Information} {Retrieval}*:347–354, 2008. ISSN: 1048776X. DOI: 10.1145/1390334.1390394. URL: http://doi.acm.org/10.1145/1390334.1390394.

[21] A. A. Freitas. Comprehensible Classification Models – a position paper. *ACM SIGKDD Explorations Newsletter*, 15(1):1–10, 2013. ISSN: 19310145. DOI: 10.1145/2594473.2594475. URL: http://dl.acm.org.miman.bib.bth.se/citation.cfm?id=2594475.

[22] N Friburger and D Maurel. Textual Similarity Based on Proper Names. *Proceedings of the workshop {Mathematical/Formal} Methods in Information Retrieval {(MFIR'2002)} at the 25 {thACM} {SIGIR} Conference*:155–167, 2002.

[23] S. Fu, J. Zhao, W. Cui, and H. Qu. Visual Analysis of MOOC Forums with iForum. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):201–210, 2017. ISSN: 10772626. DOI: 10.1109/TVCG.2016.2598444.

[24] V. Gold, C. Rohrdantz, and M. El-Assady. Exploratory Text Analysis using Lexical Episode Plots. *Eurographics Conference on Visualization (EuroVis) - Short Papers*, 2015. DOI: 10.2312/eurovisshort.20151130. URL: https://diglib.eg.org:443/handle/10.2312/eurovisshort.20151130.085-089%5Cnhttp://bib.dbvis.de/uploadedFiles/085089.pdf.

[25] R. Herbrich, T. Graepel, and K. Obermayer. Support Vector Learning for Ordinal Regression. *Science*, 1:97–102, 1999. ISSN: 05379989. DOI: 10.1049/cp:19991091. URL: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=819548.

[26] E. Hoque and G. Carenini. ConVis: A visual text analytic system for exploring blog conversations. *Computer Graphics Forum*, 33(3):221–230, 2014. ISSN: 14678659. DOI: 10.1111/cgf.12378.

[27] M Hossin and M. N. Sulaiman. a Review on Evaluation Metrics for Data Classification Evaluations. *International Journal of Data Mining & Knowledge Management Process (IJDKP)*, 5(2):1–11, 2015. ISSN: 2231-007X. DOI: 10.5121/ijdkp.2015.5201.

[28] J. Huang, M. Zhou, and D. Yang. Extracting chatbot knowledge from online discussion forums. *IJCAI International Joint Conference on Artificial Intelligence*:423–428, 2007. ISSN: 10450823.

[29] Y.-M. Huang and S.-X. Du. Weighted support vector machine for classification with uneven training class sizes. In *Machine Learning and Cybernetics, 2005. Proceedings of 2005 International Conference on*, volume 7, pages 4365–4369. IEEE, 2005.

[30] S. Jänicke, G. Franzini, M. F. Cheema, and G. Scheuermann. On close and distant reading in digital humanities: a survey and future challenges. *Proc. EuroVis, Cagliari, Italy*, 2015.

[31] N. Japkowicz. The Class Imbalance Problem: Significance and Strategies. *Proceedings of the 2000 International Conference on Artificial Intelligence*:111–117, 2000. DOI: 10.1.1.35.1693.

[32] K. J. Keim Daniel, G. r. E. Mansmann, and Florian. *Mastering the Information Age Solving Problems with Visual Analytics*. 2010, pages 57 –86. ISBN: 978-3-905673-77-7. DOI: 10.1016/j.procs.2011.12.035. URL: http://diglib.eg.org.

[33] B. Kerr. Thread Arcs: An email thread visualization. *Proceedings - IEEE Symposium on Information Visualization, INFO VIS*, 22850:211–218, 2003. ISSN: 1522404X. DOI: 10.1109/INFVIS.2003.1249028.

[34] J. Krause, A. Dasgupta, J. Swartz, Y. Aphinyanaphongs, and E. Bertini. A Workflow for Visual Diagnostics of Binary Classifiers using Instance-Level Explanations. *IEEE Conference on Visual Analytics Science and Technology*:1–11, 2017. URL: http://arxiv.org/abs/1705.01968.

[35] V. Kumar and R. Sridhar. Unsupervised Topic Modeling for Short Texts Using Distributed Representations of Words. *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*:192–200, 2015. DOI: 10.3115/v1/W15-1526.

[36] H. Lee, M. Recasens, A. Chang, M. Surdeanu, and D. Jurafsky. Joint Entity and Event Coreference Resolution across Documents. *(EMNLP-CoNLL 2012) Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, (July):489–500, 2012. URL: http://www.aclweb.org/anthology/D12-1045.

[37] C. Lin, J.-M. Yang, R. Cai, X.-J. Wang, and W. Wang. Simultaneously Modeling Semantics and Structure of Threaded Discussions: A Sparse Coding Approach and Its Applications. *Proceedings of the 32Nd International ACM SIGIR Conference on Research and Development in Information Retrieval*:131–138, 2009. DOI: 10.1145/1571941.1571966.

[38] S. Liu, J. Yin, X. Wang, W. Cui, K. Cao, and J. Pei. Online visual analytics of text streams. *IEEE Transactions on Visualization and Computer Graphics*, 22(11):2451–2466, 2016. ISSN: 10772626. DOI: 10.1109/TVCG.2015.2509990.

[39] Y. Liu, F. Chen, and Y. Chen. Learning thread reply structure on patient forums. *Proceedings of the 2013 international workshop on Data management & analytics for healthcare - DARE '13*:1–4, 2013. DOI: 10.1145/2512410.2512426. URL: http://dl.acm.org/citation.cfm?doid=2512410.2512426.

[40] Y Lou, R Caruana, and J Gehrke. Intelligible models for classification and regression. *KDD '12: Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*:1–9, 2012. DOI: 10.1145/2339530.2339556. URL: http://dl.acm.org/citation.cfm?id=2339556%5Cnpapers2://publication/uuid/E902B70F-D025-48C7-9E7F-DA86956BFE92.

[41] D Martens, B Baesens, T. Van Gestel, and J. Vanthienen. Comprehensible credit scoring models using rule extraction from support vector machines. *European Journal of Operational Research*, 183(13):1466–1476, 2007. ISSN: 0377-2217. DOI: 10.1016/j.ejor.2006.04.051.

[42] D. Martens, J. Huysmans, R. Setiono, J. Vanthienen, and B. Baesens. Rule extraction from support vector machines: an overview of issues and application in credit scoring. *Rule Extraction from Support Vector Machines*, 63(2008):33–63, 2008. DOI: 10.1007/978-3-540-75390-2{_}2.

[43] S. Mizzaro. How many relevances in information retrieval? *Interacting with Computers*, 10(3):303–320, 1998. ISSN: 09535438. DOI: 10.1016/S0953-5438(98)00012-5.

[44] D. POWERS. Evaluation: From Precision, Recall and F-Measure To Roc, Informedness, Markedness & Correlation. *Journal of Machine Learning Technologies*, 2(1):37–63, 2011. ISSN: 2229-3981. DOI: 10.1.1.214.9232. URL: http://www.bioinfopublication.org/files/articles/2_1_1_JMLT.pdf.

[45] X.-H. Phan, L.-M. Nguyen, and S. Horiguchi. Learning to Classify Short and Sparse Text & Web with Hidden Topics from Large-scale Data Collections. *Proceeding of the 17th international conference on World Wide Web - WWW '08*:91–100, 2008. ISSN: 08885885. DOI: 10.1145/1367497.1367510. URL: http://portal.acm.org/citation.cfm?doid=1367497.1367510.

[46] D. M. W. Powers. The problem with kappa. *Conference of the European Chapter of the Association for Computational Linguistics*:345–355, 2012.

[47] T. R. Prajwala. A Comparative Study on Decision Tree and Random Forest Using R Tool. *International Journal of Advanced Research in Computer and Communication Engineering*, 4(1):196–199, 2015. ISSN: 22781021. DOI: 10.17148/IJARCCE.2015.4142. URL: http://ijarcce.com/upload/2015/january/IJARCCE3L.pdf.

[48] P. Refaeilzadeh, L. Tang, and H. Liu. Cross-validation. In *Encyclopedia of database systems*, pages 532–538. Springer, 2009.

[49] S. Robertson, S Walker, M Hancock-Beaulieu, and M Gatford. Okapi in trec-3, text retrieval conference trec-3, us national institute of standards and technology, gaithersburg, usa. *NIST Special Publication*:500–225, 1994.

[50] M. Rosen-Zvi, T. Griffiths, M. Steyvers, and P. Smyth. The Author-Topic Model for Authors and Documents, 2003. ISSN: 01689002. DOI: 10.1016/j.nima.2010.11.062. URL: https://mimno.infosci.cornell.edu/info6150/readings/398.pdf.

[51] A. Schuth, M. Marx, and M. d. Rijke. Extracting the discussion structure in comments on news-articles. *Proceedings of the 9th annual ACM international workshop on Web information and data management*:97–104, 2007. DOI: 10.1145/1316902.1316919.

[52] J. Seo, W. B. Croft, and D. a. Smith. Online community search using thread structure. *Conference on Information and Knowledge Management*:1907–1910, 2009. DOI: 10.1145/1645953.1646262. URL: http://maroo.cs.umass.edu/pub/web/getpdf.php?id=866%5Cnhttp://portal.acm.org/citation.cfm?doid=1645953.1646262.

[53] H. Sharma and S. Kumar. A Survey on Decision Tree Algorithms of Classification in Data Mining. *International Journal of Science and Research*, 5(4):2094–2097, 2016.

[54] B. Shneiderman. The eyes have it: a task by data type taxonomy for information visualizations. *Proceedings 1996 IEEE Symposium on Visual Languages*:336–343, 1996. ISSN: 1049-2615. DOI: 10.1109/VL.1996.545307. URL: http://ieeexplore.ieee.org/document/545307/.

[55] A. Singhal. Modern Information Retrieval: A Brief Overview. *Bulletin of the Ieee Computer Society Technical Committee on Data Engineering*, 24(4):1–9, 2001. ISSN: 00218979. DOI: 10.1.1.117.7676âĂŐ. URL: http://160592857366.free.fr/joe/ebooks/ShareData/ModernInformationRetrieval-ABriefOverview.pdf.

[56] I. Soboroff, N. Craswell, and A. P. d. Vries. Overview of the TREC 2005 Enterprise Track. *Trec*, 5:199–205, 2006. ISSN: 1048776X.

[57] B. Soediono. The Handbook of Brain Theory and Neural Networks. *Journal of Chemical Information and Modeling*, 53:719–725, 1989. ISSN: 1098-6596. DOI: 10.1017/CBO9781107415324.004.

[58] K. Spärck Jones. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 28(1):11–21, 1972.

[59] C. Staelin. Parameter selection for support vector machines. *Hewlett-Packard Company, HPL-2002-354 (R.1)*, 354:1–4, 2003. ISSN: 1873-4324. DOI: 10.1016/j.aca.2011.07.027.

[60] C. D. Stolper, A. Perer, and D. Gotz. Progressive visual analytics: User-driven visual exploration of in-progress analytics. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):1653–1662, 2014. ISSN: 10772626. DOI: 10.1109/TVCG.2014.2346574.

[61] S. A. Tabrizi, A. Shakery, M. Asadpour, M. Abbasi, and M. A. Tavallaie. Personalized PageRank Clustering: A graph clustering algorithm based on random walks. *Physica A: Statistical Mechanics and its Applications*, 392(22):5772–5785, 2013. ISSN: 03784371. DOI: 10.1016/j.physa.2013.07.021. URL: http://dx.doi.org/10.1016/j.physa.2013.07.021.

[62] M Trampuš and M Grobelnik. Visualization of online discussion forums. *Workshop on Pattern Analysis Applications*, 11:134–141, 2010. URL: http://jmlr.org/proceedings/papers/v11/trampus10a/trampus10a.pdf.

[63] A. Vellido, J. D. Martin-Guerroro, and P Lisboa. Making machine learning models interpretable. *20th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN)*, (April):163–172, 2012.

[64] L. b. Wang, M. b. Lui, S. b. Kim, J. Nivre, and T. b. Baldwin. Predicting thread discourse structure over technical web forums. *EMNLP 2011 - Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*:13–25, 2011. URL: http://www.scopus.com/inward/record.url?eid=2-s2.0-80053286100&partnerID=40&md5=9d79293e8801989ec220c8375bdc5952.

[65] L. Wang and D. W. Oard. Context-based message expansion for disentanglement of interleaved text conversations. *NAACL '09: Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, (June):200–208, 2009. DOI: 10.3115/1620754.1620783.

[66] Y.-C. Wang, M. Joshi, W. W. Cohen, and C. Rosé. Recovering Implicit Thread Structure in Newsgroup Style Conversations. *Artificial Intelligence*:152–160, 2007.

[67] F. Wanner and D. a. Keim. ForA Vis - Explorative User Forum Analysis. (14), 2011.

[68] G. Weiss, K McCarthy, and B. Zabar. Cost-sensitive learning vs. sampling: Which is best for handling unbalanced classes with unequal error costs? *Dmin*:1–7, 2007. URL: http://storm.cis.fordham.edu/~gweiss/papers/dmin07-weiss.pdf.

[69] W. Xi, J. Lind, and E. Brill. Learning effective ranking functions for newsgroup search. *Proceedings of the 27th annual international ACM-SIGIR Conference on Research and Development in Information Retrieval*:394–401, 2004. DOI: 10.1145/1008992.1009060. URL: http://dl.acm.org/citation.cfm?id=1009060.

[70] X. Yan, J. Guo, Y. Lan, and X. Cheng. A biterm topic model for short texts. *WWW '13 Proceedings of the 22nd international conference on World Wide Web*:1445–1456, 2013. ISSN: 145032035X. DOI: 10.1145/2488388.2488514. URL: http://dl.acm.org/citation.cfm?id=2488388.2488514.

[71] J.-Y. Yeh and A. Harnly. Email thread reassembly using similarity matching. *Third Conference on Email and Anti-Spam (CEAS)*:64–71, 2006. URL: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.89.8617&rep=rep1&type=pdf.