Headerless, Quoteless, but not Hopeless? Using Pairwise Email Classification to Disentangle Email Threads

Emily K. Jamison [‡] **and Iryna Gurevych** [†‡]
† Ubiquitous Knowledge Processing Lab (UKP-DIPF),
German Institute for Educational Research and Educational Information
Schloßstr. 29, 60486 Frankfurt, Germany
[‡]Ubiquitous Knowledge Processing Lab (UKP-TUDA),
Department of Computer Science, Technische Universität Darmstadt
Hochschulstr. 10, 64289 Darmstadt, Germany
http://www.ukp.tu-darmstadt.de

Abstract

Thread disentanglement is the task of separating out conversations whose thread structure is implicit, distorted, or lost. In this paper, we perform email thread disentanglement through pairwise classification, using text similarity measures on non-quoted texts in emails. We show that i) content text similarity metrics outperform style and structure text similarity metrics in both a class-balanced and class-imbalanced setting, and ii) although feature performance is dependent on the semantic similarity of the corpus, content features are still effective even when controlling for semantic similarity. We make available the Enron Threads Corpus, a newly-extracted corpus of 70,178 multi-email threads with emails from the Enron Email Corpus.

1 Introduction

Law enforcement agencies frequently obtain large amounts of electronic messages, such as emails, which they must search for evidence. However, individual messages may be useless without the conversational context they occur in. Most modern emails contain useful metadata such as the MIME header `In-Reply-To`, which marks relations between emails in a thread and can be used to disentangle threads. However, there are easy methods of obfuscating email threads: opening an email account for a single purpose; using multiple email accounts for one person; sharing one email account among multiple persons; changing the `Subject` header; and removing quoted material from earlier in the thread.

How can emails be organized by thread without metadata such as their MIME headers?

We propose to use text similarity metrics to identify emails belonging to the same thread. In this paper, as a first step for temporal thread disentanglement, we perform pairwise classification experiments on texts in emails using no MIME headers or quoted previous emails. We have found that content-based text similarity metrics outperform a Dice baseline, and that structural and style text similarity features do not; adding these latter feature groups does not significantly improve total performance. We also found that content-based features continue to outperform the others in both a class-balanced and class-imbalanced setting, as well as with semantically controlled or non-controlled negative instances.

In NLP, Elsner and Charniak (2010) described the task of *thread disentanglement* as "the clustering task of dividing a transcript into a set of distinct conversations," in which extrinsic thread delimitation is unavailable and the threads must be disentangled using only intrinsic information. In addition to emails with missing or incorrect MIME headers, entangled electronic conversations occur in environments such as interspersed Internet Relay Chat conversations, web 2.0 article response conversations that do not have a hierarchical display order, and misplaced comments in Wiki Talk discussions.

Research on disentanglement of conversation threads has been done on internet relay chats (Elsner and Charniak, 2010), audio chats (Aoki et al., 2003), and emails *with* headers and quoted material (Yeh, 2006; Erera and Carmel, 2008). However, to the best of our knowledge, no work has investigated reassembling email threads *without* the help of MIME headers or quoted previous emails.

Proceedings of Recent Advances in Natural Language Processing, pages 327–335,
Hissar, Bulgaria, 7-13 September 2013.

Previous researchers have used a number of email corpora with high-precision (non-`Subject`-clustered) thread marking. Joti et al. (2010) used the BC3 corpus of 40 email threads and 3222 emails for topic segmentation. Carenini et al. (2008) annotated 39 email "conversations" from the Enron Email Corpus for email summarization. Wan and McKeown (2004) used a privately-available corpus of 300 threads for summary generation. Rambow et al. (2004) used a privately-available corpus of 96 email threads for thread summarization.

2 Data

The Enron Email Corpus (EEC)[1] consists of the 517,424 emails (159 users' accounts and 19,675 total senders) that existed on the Enron Corporation's email server (i.e., other emails had been previously deleted, etc) when it was made public .

2.1 Gold Standard Thread Extraction from the Enron Email Corpus

We define an email thread as a directed graph of emails connected by *Reply* and *Forward* relations. In this way, we attempt to identify email discussions between users. However, the precise definition of an email thread actually depends on the implementation that we, or any other researchers, used to identify the thread.

Previous researchers have derived email thread structure from a variety of sources. Wu and Oard (2005), and Zhu et al. (2005) auto-threaded all messages with identical, non-trivial, `Fwd:` and `Re:`-stripped `Subject` headers. Klimt and Yang (2004) auto-threaded messages that had stripped `Subject` headers and were among the same users (addresses). Lewis and Knowles (1997) assigned emails to threads by matching quotation structures between emails. Wan and McKeown (2004) reconstructed threads by header Message-ID information. Rambow et al. (2004) used a privately-available corpus of 96 email threads, but did not specify how they determined the threads.

As the emails in the EEC do not contain any inherent thread structure, it was necessary for us to create email threads. First, we implemented Klimt and Yang (2004)'s technique of clustering the emails into threads that have the same `Subject` header (after it has been stripped of prefixes such as `Re:` and `Fwd:`) and shared participants. To determine whether emails were among the same users, we split a `Subject`-created email proto-thread apart into any necessary threads, such that the split threads had no senders or recipients (including `To`, `CC`, and `BCC`) in common.

The resulting email clusters had a number of problems. Clusters tended to over-group, because a single user included as a recipient for two different threads with the `Subject` "Monday Meeting" would cause the threads to be merged into a single cluster. In addition, many clusters consisted of all of the issues of a monthly subscription newsletter, or nearly identical petitions (see Klimt and Yang (2004)'s description of the "Demand Ken Lay Donate Proceeds from Enron Stock Sales" thread), or an auto-generated log of Enron computer network problems auto-emailed to the Enron employees in charge of the network. Such clusters of "broadcast" emails do not satisfy our goal of identifying email discussions between users.

Many email discussions between users exist in previously quoted emails auto-copied at the bottom of latter emails of the thread. A single-annotator hand-investigation of 465 previously quoted emails from 20 threads showed that none of them had interspersed comments or had otherwise been altered by more recent thread contributors. Threads in the EEC are quoted multiple times at various points in the conversation in multiple surviving emails. In order to avoid creating redundant threads, which would be an information leak risk during evaluation, we selected as the thread source the email from each Klimt and Yang (2004) cluster with the most quoted emails, and discarded all other emails in the cluster. We used the quote-identifying regular expressions from Yeh (2006) (see Table 1) to identify quoted previous emails.[2]

There are two important benefits to the creation methodology of the Enron Threads Corpus[3]. First, since the emails were extracted from the same document, and the emails would only have been included in the same document by the email client if one was a `Reply` or `Forward` of the other, precision is very high (approaching 100%).[4] This is

[1] The EEC is in the public domain: `http://www.cs.cmu.edu/~enron/`

[2] The variety of email clients used at the time of these emails results in a variety of headers available in the EEC. Also, some emails have no sender, etc., because they were only saved as incomplete drafts.

[3] We have made the Enron Threads Corpus available online at `www.ukp.tu-darmstadt.de/data/text-similarity/email-disentanglement`

[4] In a handcount of 465 emails and 20 email threads, our

[-]+ Auto forwarded by <anything >[-]+
[-]+ Begin forwarded message [-]+
[-]+ cc:Mail Forwarded [-]+
[-]+ Forwarded by <person >on <datetime >[-]+
[_]+ Forward Header [_]+
[-]+ Forwarded Letter [-]+
[-]+ Forwarded Message: [-]+
"<person >" wrote:
Starts with To:
Starts with <
... and more ...

Table 1: Representative examples of Yeh (2006) regular expressions for identifying quoted emails.

Thread Size	Num threads
2	40,492
3	15,337
4	6,934
5	3,176
6	1,639
7	845
8	503
9	318
10	186
11-20	567
21+	181

Table 2: Thread sizes in the Enron Threads Corpus.

better precision than threads clustered from separate email documents, which may have the same `Subject`, etc. generating false positives. Some emails will inevitably be left out of the thread, reducing recall, because they were not part of the thread branch that was eventually used to represent the thread, or simply because they were not quoted. Our pairwise classification experiments, described in Section 4, are unaffected by this reduced recall, because each experimental instance includes only a pair of emails, and not the entire thread.

Second, because the thread source did not require human annotation, using quoted emails gives us an unprecedented number of threads as data: 209,063 emails in 70,178 threads of two emails or larger. The sizes of email threads in the Enron Threads Corpus is shown in Table 2. Emails have an average of 80.0±201.2 tokens, and an average count of 4.4±9.3 sentences. Many of the emails are quite short: 18% are under 10 tokens, 19% are 10-20 tokens, and 13% are 20-30 tokens.

3 Text Similarity Features

We cast email thread disentanglement as a text similarity problem. Ideally, there exists a text similarity measure that marks pairs of emails from the same thread as *more similar* than pairs of emails from different threads. We evaluate a number of text similarity measures, divided according to Bär et al. (2011)'s three groups: Content Similarity, Structural Similarity, Style Similarity. Each set of features investigates a different manner in which email pairs from the same thread may be identified. In our experiments, all features are derived from the body of the email, while all headers such as `Recipients`, `Subject`, and `Timestamp` are ignored.

Content features. Content similarity metrics capture the string overlap between emails with similar content. A pair of emails with a high content overlap is shown below.

The *Longest Common Substring measure* (Gusfield, 1997) identifies uninterrupted common strings, while the *Longest Common Subsequence measure* (Allison and Dix, 1986) and the single-text-length-normalized *Longest Common Subsequence Norm measure* identify common strings containing interruptions and text replacements and *Greedy String Tiling measure* (Wise, 1996) allows reordering of the subsequences. Other measures which treat texts as sequences of characters and compute similarities with various metrics include *Levenshtein* (1966), *Monge Elkan Second String measure* (Monge and Elkan, 1997), *Jaro Second String* measure (Jaro, 1989), and *Jaro Winkler Second String* measure (Winkler, 1990). A *Cosine Similarity-type measure* was used, based on term frequency within the document. Sets of n-grams from the two emails are compared using the Jaccard coefficient (from Lyon et al. (2004)) and Broder's (1997) *Containment* measure.

Structural features. Structural features attempt to identify similar syntactic patterns between the two texts, while overlooking topic-specific vocabulary. We propose that sturctural features, as well as style features below, may help in classification by means of communication accommodation theory (Giles and Ogay, 2007).

Stamatatos's *Stopword n-grams* (2011) capture syntactic similarities, by identifying text reuse where just the content words have been replaced and the stopwords remain the same. We measured the stopword n-gram overlap with Broder's (1997) *Containment* measure and four different stopword lists. We also tried the *Containment* measure and an *NGram Jaccard measure* with *part-of-speech* tags. *Token Pair Order* (Hatzivassiloglou et al.

system misidentified about 1% of emails from regular expression error.

1999) uses pairs of words occurring in the same order for the two emails; *Token Pair Distance* (Hatzivassiloglou et al., 1999) measures the distance between pairs of words. Both measures use computed feature vectors for both emails along all shared word pairs, and the vectors are compared with Pearson correlation.

Style features. Style similarity reflects authorship attribution and surface-level statistical properties of texts.

Type Token Ratio (TTR) measure calculates text-length-sensitive and text-homogeneity-sensitive vocabulary richness (Templin, 1957). However, as this measure is sensitive to differences in document length between the pair of documents (documents become less lexically diverse as length and token count increases but type count levels off), and fluctuating lexical diversity as rhetorical strategies shift within a single document, we also used *Sequential TTR* (McCarthy and Jarvis, 2010), which corrects for these problems. *Sentence Length* and *Token Length* (inspired by (Yule, 1939)) measure the average number of tokens per sentence and characters per token, respectively. *Sentence Ratio* and *Token Ratio* compare *Sentence Length* and *Token Length* between the two emails (Bär et al., 2011). *Function Word Frequencies* is a Pearson's correlation between feature vectors of the frequencies of 70 pre-identified function words from Mosteller and Wallace (1964) across the two emails. We also compute *Case Combined Ratio*, showing the percentage of UPPERCASE characters in both emails combined ($\frac{UPPERCASE_{e1}+UPPERCASE_{e2}}{ALLCHARS_{e1}+ALLCHARS_{e2}}$), and *Case Document similarity*, showing the similarity between the percentage of UPPERCASE characters in one email versus the other email.

4 Evaluation

In this series of experiments, we evaluate the effectiveness of different feature groups to classify pairs of emails as being from the same thread (*positive*) or not (*negative*). Each instance to be classified is represented by the features from a pair of emails and the instance classification, positive or negative.

We used a variation of K-fold cross-validation for evaluation. The 10 folds contained carefully distributed email pairs such that email pairs with emails from the same thread were never used in pairs of training, development, and testing sets, to avoid information leakage. All instances were at one point in a test set. Instance division was roughly 80% training, 10% development, and 10% test data. Reported results are the weighted averages across all folds.

The evaluation used logistic regression, as implemented in Weka (Hall et al., 2009). Default parameters were used. We use a baseline algorithm of Dice Similarity between the texts of the two emails as a simple measure of set similarity. We created an upper bound by annotating 100 positive and 100 negative instances. A single native English speaker annotator answered the question, "Are these emails from the same thread?"

4.1 Data Sampling

Although we had 413,814 positive instances available in the Enron Threads Corpus, we found that classifier performance was unaffected by the amount of training data, down to very low levels (see Figure 1). However, because the standard deviation in the data did not level out until around 1,200 class-balanced training instances[5], we used this number of positive instances (600) in each of our experiments.

In order to estimate effectiveness of features for different data distributions, we used three different subsampled datasets.

Random Balanced (RB) Dataset. The first dataset is class-balanced and uses 1200 training instances. Minimum email length is one word. For every positive instance we used, we created a negative email pair by taking the first email from the positive pair and pseudo-randomly pairing it with another email from a different thread that was assigned to the same training, development, or test set.

However, the probability of semantic similarity between two emails in a positive instance is much greater than the probability of semantic similarity between two emails in a randomly-created negative instance. The results of experiments on our first dataset reflect both the success of our text similarity metrics and the semantic similarity (i.e., topical distribution) within our dataset. The topical distribution will vary immensely between different email corpora. To investigate the performance of our features in a more generaliable environment, we created a subsample dataset that con-

[5]Each fold used 1,200 training instances and 150 test instances.

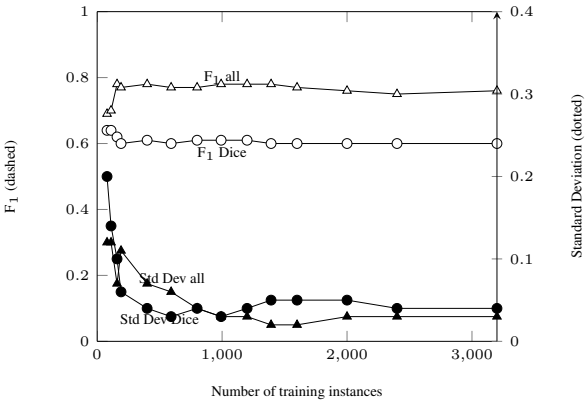

Figure 1: Training data sizes and corresponding F_1 and standard deviation.

trolls for semantic similarity within and outside of the email thread.

Semantically Balanced (SB) Dataset. The second dataset combines the same positive instances as the first set with an equal number of semantically-matched negative instances for a training size of 1200 instances, and a minimum email length of one word. For each positive instance, we measured the semantic similarity within the email pair using Cosine Similarity and then created a negative instance with the same ($\pm$.005) similarity. Emails had an average of 96$\pm$287 tokens and 5$\pm$11sentences, and a similar token size distribution as SB.

Random Imbalanced (RI) Dataset. However, both the RB and SB datasets use a class-balanced distribution. To see if our features are still effective in a class-imbalanced environment, we created a third dataset with a 90% negative, 10% positive distribution for both the training and test sets[6]. Specifically, we used the first dataset and then added an extra 8 negative instances for each positive instance. Experiments with this dataset use 10-fold cross validation, where each fold has 6000 training and 750 test instances. No minimum email length was used, similar to a more natural distribution.

4.2 Results

Our results are shown in Table 3. Since we aim to detect pairs of emails belonging to the same thread rather than unrelated emails, we measure the system performance on the positive class. We use the standard F-measure of $F_1=\frac{2\times P(pos)\times R(pos)}{P(pos)+R(pos)}$. As a measure to show performance on both positive and negative classes, we provide a standard accuracy measure of Acc=$\frac{TP+TN}{TP+FN+TN+FP}$. Feature groups are shown in isolation as well as the complete set of features minus one group. [7]

With the RB corpus, the best performing single feature configuration, content features group (P=.83 $\pm$.04), matches the human upper bound precision(P=.84). The benefit of content features is confirmed by the reductions in complete feature set performance when they are left out. The content features group was the only group to perform significantly above the Dice baseline. Adding the other feature groups does not significantly improve the overall results. Further leave-one-out experiments revealed no single high performing feature within the content features group.

Structural features produced low performance, failing to beat the Chance baseline. Structural similarity from rhetorical strategy is rare in an email conversational setting. Any structural benefits are likely to come from sources unavailable in a disguised email situation, such as auto-signatures identifying senders as the same person. The low results on structural features show that we are not relying on such artifacts for classification.

Style features were also unhelpful, failing to significantly beat the Dice baseline. The features failed to identify communication accomodation within the thread.

Results on the SB dataset show that there is a noticeable drop in classification for all feature groups when negative instances have a similar semantic similarity as positive instances. The configuration with all features showed a 15 percentage point drop in precision, and a 12 percentage point drop in accuracy. However, content features continues to be the best performing feature group with semantically similar negative instances, as with random negative instances, and outperformed the Dice baseline. Adding the additional feature groups does not significantly improve overall performance.

The results on the RI corpus mirror results from the balanced (RB) corpus. The best-performing

[6]This class imbalance is still artificially lower than a more natural 99.99+% negative natural class imbalance.

[7]Additionally, we tried a semantic similarity measures features group. We used Gabrilovich & Markovitch's (2007) Explicit Semantic Analysis (ESA) vector space model, with three different lexical-semantic resources: WordNet, Wikipedia, and Wiktionary. The performance of this feature group (P=.50) was not good enough to include in Table 3.

Feature	RB F_1	SB F_1	RI F_1	RB Acc	SB Acc	RI Acc
Chance	.50	.50	.90	.50	.50	.90
Dice Baseline	.61 ±.04	.56 ±.04	.09 ±.04	.63 ±.03	.58 ±.03	.9 ±.0
Upper Bound	.89	-	-	.89	-	-
Just content	.78 ±.03	.65 ±.04	.38 ±.06	.79 ±.03	.67 ±.03	.92 ±.01
Just struct	.42 ±.05	.33 ±.04	.06 ±.05	.55 ±.03	.52 ±.03	.90 ±.00
Just style	.60 ±.05	.57 ±.03	.00 ±.00	.60 ±.04	.56 ±.03	.90 ±.00
No content	.60 ±.03	.55 ±.03	.08 ±.05	.62 ±.03	.57 ±.02	.90 ±.00
No struct	.78 ±.03	.66 ±.03	.41 ±.06	.79 ±.02	.67 ±.02	.92 ±.01
No style	.78 ±.03	.63 ±.04	.38 ±.06	.79 ±.03	.65 ±.03	.92 ±.00
Everything	.78 ±.02	.65 ±.03	.40 ±.05	.79 ±.02	.67 ±.03	.92 ±.00

Table 3: Email pair classification results, with random negative instances.

Text	Freq in Corpus
FYI	48
FYI <*name* >	23
one person's autosignature	7
Thanks!	5
Please print.	5
yes	4
FYI, Kim.	3
ok	3
please handle	3

Table 4: Common texts and their frequencies in the corpus.

individual feature group in both experiments was the content feature group; in the class-imbalanced experiments the group alone beats the Dice baseline in F_1 by 29 percentage points and reduces accuracy error by about 20%.

Elsner and Charniak (2011) use coherence models to disentangle chat, using some features (entity grid, topical entity grid) which correspond to the information in our content features group. They also found these content-based features to be helpful.

4.3 Inherent limitations

Certain limitations are inherent in email thread disentanglement. Some email thread relations cannot be detected with text similarity metrics, and require extensive discourse knowledge, such as the emails below.

> **Email1:** *Can you attend the Directors Fund Equity Board Meeting next Wednesday, Nov 5, at 3pm?*
>
> **Email2:** *Yes, I will be there.*

Several other problems in email thread disentanglement cannot be solved with any discourse knowledge. One problem is that some emails are identical or near-identical; there is no way to choose between textually identical emails. Table 4 shows some of the most common email texts in our corpus, based on a $<.05$ similarity value from Jaro Second String similarity, as described in Section 3.

However, near identical texts make up only a small portion of the emails in our corpus. In a sample of 5,296 emails, only 3.6% of email texts were within a .05 Jaro Second String similarity value of another text.

Another problem is that some emails are impossible to distinguish without world and domain knowledge. Consider a building with two meeting rooms: *A101* and *A201*. Sometimes *A101* is used, and sometimes *A201* is used. In response to the question, *Which room is Monday's meeting in?*, there may be no way to choose between *A101* and *A201* without further world knowledge.

Another problem is topic overlap. For example, in a business email corpus such as the EEC, there are numerous threads discussing Monday morning 9am meetings. The more similar the language used between threads, the more difficult the disentanglement becomes, using text similarity. This issue is addressed with the SB dataset.

Finally, our classifier cannot out-perform humans on the same task, so it is important to note human limitations in email disentanglement. Our human upper bound is shown in Table 3. We will further address this issue in Sections 4.4.

4.4 Error Analysis

We inspected 50 email pairs each of true positives, false positives, false negatives, and true negatives from our RB experiments[8] . We inspected for both technical details likely to affect classification, and for linguistic features to guide future research. Technical details included small and large text errors (such as unidentified email headers or incorrect email segmentation), custom and non-custom email signatures, and the presense of large signatures likely to affect classification. Linguistic features included an appearance of consecutivity (emails appear in a Q/A relation, or one is informative and one is 'please print', etc.), similarity of social style ("Language vocab level, professionalism, and social address are a reasonable match"), and the annotator's perception that the emails could be from the same thread.

An example of a text error is shown below.

> **Sample text error:**
>
> *Craig Young*
>
> *09/08/2000 01:06 PM*

[8]Despite the semantic similarity control, an error analysis of our SB experiments showed no particularly different results.

Names and dates occur frequently in legitimate email text, such as meeting attendance lists, etc., which makes them difficult to screen out. Emails from false positives were less likely to contain these small errors (3% versus 14%), which implies that the noise introduced from the extra text has more impact than the false similarity potentially generated by similar text errors. Large text errors (such as 2 emails labelled as one) occurred in only 1% of emails and were too rare to correlate with results.

Autosignatures, such as the examples below, mildly impacted classification.

Custom Autosignature:
Carolyn M. Campbell
713-276-7307 (phone)

Non-custom Autosignature:
Get your FREE download of MSN Explorer
at http://explorer.msn.com

Instances classified as negative (both FN and TN) were marginally more likely to have had one email with a non-customized autosignature (3% versus 1.5%) or a customized auto-signature (6.5% versus 3.5%). Autosignatures were also judged likely to affect similarity values more often on instances classified as negative (20% of instances). The presence of the autosignature may have introduced enough noise for the classifier to decide the emails were not similar enough to be from the same thread. We define a non-custom auto-signature as any automatically-added text at the bottom of the email. We did not see enough instances where both emails had an autosignature to evaluate whether similarities in autosignatures (such as a common area code) impacted results.

Some email pair similarities, observable by humans, are not being captured by our text similarity features. Nearly all (98%) positive instances were recognized by the annotator as potential consecutive emails within a thread, or non-consecutive emails but still from the same thread, whereas only 46% of negative instances were similarly (falsely) noted. Only 2% of negative instances were judged to look like they were consecutive emails within the same thread.

The following TP instance shows emails that look like they could be from the same thread but do not look consecutive.

Email1: *give me the explanations and i will think about it*
Email2: *what do you mean, you are worth it for one day*

Below is a TN instance with emails that look like they could be from the same thread but do not look consecutive.

Email1: *i do but i havent heard from you either, how are things with wade*
Email2: *rumor has it that a press conference will take place at 4:00 - more money in, lower conversion rate.*

The level of professionalism ("Language vocab level, professionalism, and social address are a reasonable match") was also notable between class categories. All TP instances were judged to have a professionalism match, as well as 94% of FN's. However, only 64% of FP's and 56% of TN's were judged to have a professionalism match. Based on a review of our misclassified instances, we are surprised that our classifier did not learn a better model based on style features (F_1=.60). Participants in an email thread appear to echo the style of emails they reply to. For instance, short, casual, all-lowercase emails are frequently responded to in a similar manner.

5 Conclusion

In this paper, we have described the creation of the Enron Threads Corpus, which we made available online. We have investigated the use of text similarity features for the pairwise classification of emails for thread disentanglement. We have found that content similarity features are more effective than style or structural features across class-balanced and class-imbalanced environments. There appear to be more stylistic features uncaptured by our similarity metrics, which humans access for performing the same task. We have shown that semantic differences between corpora will impact the general effectiveness of text similarity features, but that content features remain effective.

In future work, we will investigate discourse knowledge, highly-tuned stylistic features, and other email-specific features to improve headerless, quoteless email thread disentanglement.

Acknowledgments

This work has been supported by the Volkswagen Foundation as part of the Lichtenberg-Professorship Program under grant No. I/82806, and by the Center for Advanced Security Research (`www.cased.de`).

References

Allison, L. and Dix, T. I. (1986). A bit-string longest-common-subsequence algorithm. *Information Processing Letters*, 23:305–310.

Aoki, Paul M. and Romaine, Matthew and Szymanski, Margaret H. and Thornton, James D. and Wilson, Daniel and Woodruff, Allison. 2003. The mad hatter's cocktail party: a social mobile audio space supporting multiple simultaneous conversations. In Gilbert Cockton and Panu Korhonen, editors, *CHI*, pages 425–432. ACM.

Bär, Daniel and Zesch, Torsten and Gurevych, Iryna. 2011. A reflective view on text similarity. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing*, pages 515–520, Sep.

Broder, A. Z. (1997). On the resemblance and containment of documents. *Proceedings of Compression and Complexity of Sequences*, pages 21–29.

Carenini, Giuseppe and Ng, Raymond T. and Zhou, Xiaodong (1997). Summarizing Emails with Conversational Cohesion and Subjectivity. *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics, June 15-20, 2008, Columbus, Ohio, USA*, pages 353–361.

Elsner, Micha and Charniak, Eugene. 2010. Disentangling chat. *Comput. Linguist.*, 36(3):389–409, September.

Elsner, Micha and Charniak, Eugene. 2011. Disentangling Chat with Local Coherence Models *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics, June, 2011, Portland, Oregon, USA*, pages 1179–1189.

Erera, Shai and Carmel, David. 2008. Conversation detection in email systems. In *Proceedings of the IR research, 30th European conference on Advances in information retrieval*, ECIR'08, pages 498–505, Berlin, Heidelberg. Springer-Verlag.

Gabrilovich, E. and Markovitch, S. (2007). Computing Semantic Relatedness using Wikipedia-based Explicit Semantic Analysis. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence*, pages 1606–1611, Hyderabad, India.

Giles, Howard and Ogay, Tania (2007). Communication Accommodation Theory. In Bryan B. Whaley and Wendy Samter, editors. *Explaining Communication: Contemporary Theories and Exemplars*, Mahwah, NJ. Lawrence Erlbaum.

Gusfield, D. (1997). *Algorithms on Strings, Trees and Sequences: Computer Science and Computational Biology*. Cambridge University Press.

Hall, Mark and Frank, Eibe and Holmes, Geoffrey and Pfahringer, Bernhard and Reutemann, Peter and Witten, Ian H. (2010). The WEKA data mining software: an update. In *SIGKDD Explor. Newsl.*, 11(1):10–18.

Hatzivassiloglou, V., Klavans, J. L., and Eskin, E. (1999). Detecting text similarity over short passages: Exploring linguistic feature combinations via machine learning. In *Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, pages 203–212, College Park, MD, USA.

Jaro, M. A. (1989). Advances in record linkage methodology as applied to the 1985 census of Tampa Florida. *Journal of the American Statistical Association*, 84(406):414–420.

Joty, Shafiq and Carenini, Giuseppe and Murray, Gabriel and Ng, Raymond T. (2010). Exploiting conversation structure in unsupervised topic segmentation for emails. *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*.

Klimt, Bryan and Yang, Yiming. 2004. Introducing the enron corpus. In *CEAS*.

Levenshtein, V. I. (1966). Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics Doklady*, 10(8):707–710.

Lewis, D. D. (1966). Threading electronic mail: a preliminary study . In *Information Processing and Management* 33(2):209–217.

Lyon, C., Barrett, R., and Malcolm, J. (2004). A theoretical basis to the automated detection of copying between texts, and its practical implementation in the Ferret plagiarism and collusion detector. In *In Plagiarism: Prevention, Practice and Policies Conference*.

McCarthy, P. M. and Jarvis, S. (2010). MTLD, vocd-D, and HD-D: A validation study of sophisticated approaches to lexical diversity assessment. *Behavior research methods*, 42(2):381–392.

Monge, A. and Elkan, C. (1997). An efficient domain-independent algorithm for detecting approximately duplicate database records. In *Proceedings of the SIGMOD Workshop on Data Mining and Knowledge Discovery*, pages 23–29, Tucson, AZ, USA.

Mosteller, F. and Wallace, D. L. (1964). *Inference and disputed authorship: The Federalist*. Addison-Wesley.

Rambow, Owen and Lokesh, Shrestha and Chen, John and Lauridsen, Chirsty. (2004). Summarizing email threads. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics (NAACL) Short Paper Section*.

Stamatatos, E. (2011). Plagiarism detection using stopword n-grams. *Journal of the American Society for Information Science and Technology*, 62(12):2512–2527.

Templin, M. C. (1957). *Certain language skills in children*. University of Minnesota Press.

Wan, Stephen and McKeown, Kathy. (2004). Generating overview summaries of ongoing email thread discussions. In *Proceedings of COLING 2004*.

Weizhong Zhu, Robert B. Allen, and Min Song. 2005. Trec 2005 enterprise track results from drexel. In Ellen M. Voorhees and Lori P. Buckland, editors, *Proceedings of the Fourteenth Text REtrieval Conference, TREC 2005, Gaithersburg, Maryland, November 15-18, 2005*, volume Special Publication 500-266. National Institute of Standards and Technology (NIST).

Winkler, W. E. (1990). String Comparator Metrics and Enhanced Decision Rules in the Fellegi-Sunter Model of Record Linkage. In *Proceedings of the Section on Survey Research Methods*, pages 354–359.

Wise, M. J. (1996). YAP3: Improved detection of similarities in computer program and other texts. In *Proceedings of the 27th SIGCSE technical symposium on Computer science education*, pages 130–134, Philadelphia, PA, USA.

Wu, Yejun and Oard, Douglas W. 2005. Indexing emails and email threads for retrieval. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '05, pages 665–666, New York, NY, USA. ACM.

Yeh, Jen-Yuan. 2006. Email thread reassembly using similarity matching. In *CEAS 2006 - The Third Conference on Email and Anti-Spam, July 27-28, 2006, Mountain View, California, USA*.

Yule, G. U. (1939). On sentence-length as a statistical characteristic of style in prose: With application to two cases of disputed authorship. *Biometrika*, 30(3/4):363–390.