CEREC: A Corpus for Entity Resolution in Email Conversations

Parag Pravin Dakle
Department of Computer Science
The University of Texas at Dallas
paragpravin.dakle@utdallas.edu

Dan I. Moldovan
Department of Computer Science
The University of Texas at Dallas
moldovan@utdallas.edu

Abstract

We present the first large scale corpus for entity resolution in email conversations (CEREC). The corpus consists of 6001 email threads from the Enron Email Corpus containing 36,448 email messages and 38,996 entity coreference chains. The annotation is carried out as a two-step process with minimal manual effort. Experiments are carried out for evaluating different features and performance of four baselines on the created corpus. For the task of mention identification and coreference resolution, a best performance of 54.1 F1 is reported, highlighting the room for improvement. An in-depth qualitative and quantitative error analysis is presented to understand the limitations of the baselines considered.

1 Introduction

Entity resolution is defined as linking referring spans of text that point to the same discourse entity by CoNLL 2012 (Pradhan et al., 2012) and MUC (Grishman and Sundheim, 1996) shared tasks. The corpora used for this task primarily consist of text from news (Pradhan et al., 2012; Cybulska and Vossen, 2014; Recasens et al., 2010; Grishman and Sundheim, 1996), web-logs and transcripted dialogs.

This research focusses on the entity resolution task for email conversations. Example 1 shows a sample email message and the corresponding entities. The boldfaced tokens represent entities and the numbers beside them represent coreference chain identifiers. An Entity is defined as an object or a group of objects in the real world and a span of text referring to an entity is called a Mention. When all mentions in a text which refer to the same real-world entity are linked together, they form a coreference chain.

Example 1. Example of entity resolution task in email conversations

Date: Mon, 17 Dec 2001 14:28:03 -0800 (PST)
From: **g..barkowsky@enron.com***(1)* To: **theresa.staab@enron.com***(2)*
Subject: RE: Final Statements and Invoices for November
X-From: **Barkowsky, Gloria G.***(1)* X-To: **Staab, Theresa***(2)*
yes, **I***(1)* 'll do this. Do **you***(2)* have anything for **Crestone and Lost Creek***(3)*?

Dakle et al. (2020) first studied entity resolution in email conversations using a small annotated corpus. Following the same task definition, this paper builds on their work and makes the following key contributions:

1. A large corpus for entity resolution in email conversations (CEREC), weakly annotated for mentions and coreference chains, is presented. Detailed corpus statistics are also discussed. The corpus will be released along with the paper[1].

[1] https://github.com/paragdakle/emailcoref

Proceedings of the 28th International Conference on Computational Linguistics, pages 339–349
Barcelona, Spain (Online), December 8-13, 2020

2. Experiments with several baseline models are carried out and their results are reported. A qualitative and quantitative error analysis of the results is presented.

The paper is organized as follows: Section 2 reviews related work done on email processing and corpora for emails and entity resolution. Section 3 describes the corpus creation process and reports statistics on the created corpus. It also explores the addition of features and experiments to evaluate the same. Section 4 presents the baseline models, experiments carried out, and the results obtained. Error analysis of the results is covered in Section 5, followed by a discussion on the problem of missing context in Section 6. Section 7 concludes the work done in this paper.

2 Related Work

Email processing has been an active research topic with the earlier works focusing on email classification (Cohen and others, 1996; Whittaker and Sidner, 1996; Brutlag and Meek, 2000; Manco et al., 2002; Klimt and Yang, 2004; Alkhereyf and Rambow, 2017). This was later followed by work on intent classification (Cohen et al., 2004), searching (Soboroff et al., 2006; Minkov et al., 2008), clustering (Huang and Mitchell, 2008), and summarization (Muresan et al., 2001; Lam, 2002; Newman and Blitzer, 2003; Nenkova and Bagga, 2004; Corston-Oliver et al., 2004; Rambow et al., 2004; Carenini et al., 2007; Ulrich et al., 2008).

One of the earliest works highlighting the challenges of the thread-like nature of email conversations was carried out by Lewis and Knowles (1997). Murakoshi et. al (2000) proposed the creation of extended contribution trees to better understand the conversation structure of email threads. The impact of coreference resolution on conversations in a threaded format was first studied by Hendrickx and Hoste (2009) using a corpus of blogs and commented news for opinion mining. Although the impact was negative, it was attributed to the poor performance of the coreference system. Coreference resolution for email conversations is an unexplored problem with, to the best of our knowledge, the only work done by Dakle et. al (2020).

Numerous corpora have been used for email processing over time. University emails (Cohen and others, 1996; Cohen et al., 2004), email users survey (Whittaker and Sidner, 1996; Brutlag and Meek, 2000), private emails (Manco et al., 2002; Corston-Oliver et al., 2004), simulated emails (Lam, 2002), and email archives (Nenkova and Bagga, 2004) are few of the initial sources for email corpora. The Enron Email Corpus (Klimt and Yang, 2004) was the first large public corpus containing emails of 150 employees of the Enron Corporation. Similarly, the Avocado Research Email Collection (Douglas Oard, 2015) consists of emails from 282 accounts of a now-defunct IT company.

The task of coreference resolution, specifically entity resolution, has received attention in the natural language research community since the 1960s with noun-phrase and pronomial resolution being the early forms of the task. Although multiple corpora released over the years contain a small fraction of telephonic speech text, only a few corpora have focused on the study of the task in a purely conversational setting. Character Identification Corpus (Chen and Choi, 2016) was the first corpus to focus on the entity-linking task in this setting. It was constructed using TV show transcripts with annotations for speakers in a multi-party conversation. Aktaş et. al (2018) used a Twitter corpus to study the performance of Stanford statistical coreference system (Clark and Manning, 2015). They evaluated a corpus with 185 threads containing 278 coreference chains and reported a mediocre performance by the model.

3 Corpus

3.1 Seed Corpus

Dakle et al. (2020) in their study on entity resolution released a small manually annotated corpus containing 46 email threads from the Enron Email Corpus[2] (Klimt and Yang, 2004). The Enron Email Corpus is a multi-lingual corpus with the majority of email threads in English. The corpus consists of email threads organized in a directory structure for each user. The annotated corpus consists of 245 email messages with 866 coreference chains containing 5,834 mentions. Each mention refers to an entity of

[2] https://www.cs.cmu.edu/~./enron/

the type PERSON, ORGANIZATION, LOCATION, and DIGITAL[3]. We use the seed corpus (SC) as the starting point[4] for this work and the underlying Enron Email Corpus as the base corpus to create CEREC. Additionally, the annotation guidelines elaborated by Dakle et al. (2020) are followed in this research.

3.2 Extraction and Filtering

The first step in creating the larger corpus is to shortlist email threads from the Enron Email Corpus. An email thread conversation is considered to be a valid conversation if it contains 4 or more email messages. However, to increase the size of this shortlisted pool of email threads, we do not restrict the scope only to email threads in the *inbox* directory. For each user, email threads in all directories except *all_documents*, *discussion_threads*, *drafts*, *deleted_items*, *sent_items*, *sent*, *_sent_mail*, and *_sent* are considered. Since, email threads in previous directories are either auto-generated, discarded, or are part of other email threads, they are omitted. A total of 9,724 email threads with a minimum of 4 email messages in each thread are obtained after including additional directories.

On obtaining the initial set of candidate email threads, the following types of email threads are manually filtered from the resulting set:

1. Duplicates: An email thread that is part of a larger email thread or is a duplicate belongs to this category. The multi-recipient nature of email conversations results in one email thread possibly being present in directories of multiple users.

2. No content: Threads in which more than half of the email messages containing no body fall in this category.

3. Invalid attachments: The Enron Email Corpus consists of email threads with inline document attachments. Some email threads contain attachments as long hexadecimal strings and hence are labeled as invalid content.

4. Non-English content: Email threads in the base Enron corpus consists of messages or text in English, Spanish, Russian, German, and French. The scope of this work being restricted to English, email threads containing text in any other language are discarded.

In addition to the above types, we also discard any email threads which overlap partially or fully with those in SC. This is done as eventually SC will be used as the test set for all experiments. After filtering from the initial set, 6144 email threads are obtained. Table 1 gives a distribution of the initial email threads in each of the filtering categories. Furthermore, the unlabelled corpus contains a total of 37,315 email messages with an average thread length of 6 email messages.

Email Thread Category	Email Thread Count
Duplicates	2,867
No content	564
Invalid attachments	75
Non-English content	54
Seed corpus overlap	20
Accepted Email Threads	6,144
Total	9,724

Table 1: Distribution of email threads per filtering category

[3]A digital entity is a media or pointer to a media which is present on some form of digital storage (Dakle et al., 2020)

[4]Only 43 email threads out of the 46 have been used in this work as 3 email threads were discarded due to their overlap with the other email threads in SC

3.3 Annotation

The annotation procedure is divided into two parts: mention annotation and coreference annotation. For both parts, pre-trained SpanBERT (Joshi et al., 2019a) variant of the model proposed by Joshi et. al (2019b)[5] is used[6]. Henceforth, we will refer to this model as VanillaSpanBERT (for additional description of the model see 4.1).

3.3.1 Mention Annotation

Given an email thread, correctly identifying spans of text which refer to an entity is the task of mention identification. Here, mention identification task is framed as identifying a single coreference chain which consists of all spans of text referring to a valid entity. A valid entity is an entity of the type PERSON, ORGANIZATION, LOCATION or DIGITAL. Consider Example 1, here the single coreference chain will be *[“g..barkowsky@enron.com”, “theresa.staab@enron.com”, “Barkowsky, Gloria G.”, “Staab, Theresa”, “I”, “you”, “Crestone and Lost Creek”]*. Framing the task in this manner helps in speeding up the annotation process as it eliminates the need to perform architectural changes and carrying out experiments to test each change.

Statistic	Value
Added Mentions	2,106
Corrected Mentions	344
Deleted Mentions	325
No-change/Predicted Mentions	12,056
Total Mentions	13,837
Precision	0.93
Recall	0.86
F1-score	0.89

Table 2: Statistics for changes done during manual correction of predictions obtained on 143 email threads.

First, a VanillaSpanBERT model is trained on SC for the mention identification task. Next, this trained model is used to obtain predictions on the unlabelled corpus. From these predictions, approximately 2% (143 email threads) are manually corrected and a training set of 94 email threads and a validation set of 49 email threads is created. Table 2 shows the count of the type of changes done during the manual correction of these 143 email threads and the corresponding precision, recall, and F1-score of the trained model. In addition to correcting the predictions, we also correct sentence boundaries for these email threads. The remaining 6,001 email threads will be referred to as mention annotated corpus (MAC). The motivation to create a training and validation set is to compare the performance of models trained on gold annotated (94 email threads) and weakly annotated (MAC) training sets, respectively. These models will be referred to as M-VanillaSpanBERT$_{94}$ and M-VanillaSpanBERT$_{6001}$ respectively. Table 3 reports the results of these two models on SC. From the results, two inferences can be drawn:

1. The model M-VanillaSpanBERT$_{6001}$ performs equally well than its counterpart trained on a gold annotated corpus. Weak annotations by definition are either incomplete or contain incorrect annotations. However, based on the correction evaluation statistics (Table 2) and experiment results, an assumption that they are gold mention annotations for obtaining weak coreference annotation can be made.

2. The performance of the model M-VanillaSpanBERT$_{6001}$ illustrates the robustness of the model to ignore the noise in the weakly annotated corpus.

[5] https://github.com/mandarjoshi90/coref

[6] Note that here pre-trained SpanBERT implies a pre-trained SpanBERT base model **not** further trained on the OntoNotes corpus

Finally, both SC and the training set containing 94 email threads are used to train a VanillaSpanBERT to obtain mention annotations on 6001 email threads, thereby further improving the quality of mention annotations.

Model	P	R	F1
M-VanillaSpanBERT$_{94}$	0.94	0.82	0.8758
M-VanillaSpanBERT$_{6001}$	0.95	0.808	0.8737

Table 3: Results of two models trained on 94 gold annotated and 6,001 weakly annotated documents respectively.

3.3.2 Coreference Annotation

Post completing mention annotation on the unlabelled corpus, the next step is to perform entity coreference annotation. For this task, an approach similar to the one undertaken for obtaining mentions annotations is used. First, a gold validation set is created to assist in understanding the training performance. A set of 34 email threads is selected from the validation set used for mention annotation. Two annotators performed annotation only on the previously gold-annotated mentions. Second, a VanillaSpanBERT model is trained on the coreference annotations of SC to obtain annotations on the MAC. Mention annotations from MAC are provided as input during the coreference annotation process. The final annotated corpus will be referred to as CEREC. Table 4 provides different corpus statistics. Although the corpus contains a large number of mention annotations, 29,600 of them have been added by the model during the coreference annotation process. In addition to this, 100,385 mentions added during the mention annotation process have not been annotated by the model in this step.

Statistic	Value
Number of email threads	6001
Number of email messages	36,448
Number of words	6,569,227
Coreference Chains	38,996
Annotated Mentions	422,081
Annotated Pronouns	145,615
Length of longest coreference chain	388
Average Length of coreference chains	84.14

Table 4: CEREC statistics

3.3.3 Environment and Hyperparameters

All mention annotation experiments are carried out using the *spanbert_base* model with a maximum segment length of 256 and on an NVIDIA GeForce GTX 1080 Ti GPU with 8 12gb cores. The base variant of the SpanBERT model trains 2x faster than the large variant only for a loss of 0.1 F1 points. On the other hand, for coreference annotations, *spanbert_large* with a maximum segment length of 512 outperforms the previous configuration by 7 F1 points. However, this large variant is trained for 10 epochs, and on the CPU due to memory constraints. The *genre* feature is also removed from all models. All remaining hyperparameters in both settings are left unchanged.

3.4 Feature Addition

Training using additional features like speaker information and genre indicators on top of coreference annotations has proved to be helpful in the past. On the same lines, we evaluate three features specific to conversational texts which have a thread-like structure.

1. Message identifier (MI): For an email thread T containing N email messages, message identifier for a token *x* belonging to message *i* (*i*$\in\{0, 1, ..., N\text{-}1\}$) is *i*.

2. Section information (SI): An email message is divided into three sections: header, body, and footer[7]. The feature assigns one of the header, body and footer classes to each token in an email message.

3. Reversing an email (REV): Reversing email messages in a thread refers to ordering the messages as per the time in the email header. This is expected to enhance the model's understanding of the conversation flow in the thread.

For the evaluation, VanillaSpanBERT is used and SC with 43 email threads is used as the training set. The validation set used during the mention annotation process is used with a 14-20 email thread split to create a validation and testing set. A single annotator was used to perform feature annotation on all 77 email threads. Table 5 reports results of experiments with permutations of all features using the CoNLL average F1 metric (described in 4.3). An embedding size of 20 is chosen to encode EI and SI for all feature addition experiments.

Feature	Avg. F1 (conll)
VanillaSpanBERT	55.57
+ MI	54.40
+ SI	**56.53**
+ REV	53.94
+ REV + MI	52.15
+ REV + SI	54.18
+ MI + SI	55.29
+ REV + MI + SI	52.94

Table 5: VanillaSpanBERT evaluation results for all permutations of additional features

Table 5 shows that the addition of SI improves the performance of the model in all scenarios. SI provides information which is useful in identification mentions used for pronoun resolution. All mentions in *To* or *Cc*, or the mention in *From* are used to resolve pronouns like *I*, *you*, *me*, *us*, etc[8].

Reversing the email thread (REV) in temporal order reduces the average F1. This disproves the hypothesis presented before. However, it is important to note that the test size for these experiments consisted of only 20 email threads. Finally, the addition of MI does not help the model. MI provides the model with message boundary information which can be used to merge inter email message clusters but fails to have a positive impact in the current setting.

4 Experiments

4.1 Baselines

Header baseline1 (Hb1): A simple baseline of resolving pronouns based on the participants in the email header is constructed. All first person singular pronouns ("I", "me", "my", "mine", "myself") are chained to the sender, and second-person pronouns ("you", "your", "yours", "yourself", "yourselves") to the recipients respectively. First-person plural pronouns ("we", "us", "our", "ours", "ourselves") are linked to both the sender and the recipients of the email message. In addition to this, all non-pronomial mentions which are the same or have overlapping words are chained together. This baseline is rule-based and does not consider the surrounding context.

Header baseline2 (Hb2): This is similar to Header baseline1 except for how first-person plural pronouns are resolved. In this baseline, all first-person plural pronouns in an email message are chained together into one coreference chain and not to the sender or recipients of that message. Furthermore, each first-person plural pronoun chain in an email message is merged with the corresponding chains in every other message of that email thread.

[7]Footer is defined as the system generated privacy notification or company advertisement. All privacy notifications have been ignored in this work.

[8]This excludes the cases when the sender or an alias of the sender is one of the recipients of the email

c2f-coref (C2F): The model proposed by Lee et. al (2018) is used for this baseline[9]. This was the first end-to-end neural coreference resolution model. It uses highway LSTMs to generate embeddings for each span and then with a span-ranking model decides which of the previous spans is a suitable antecedent (if any). The inputs to the LSTMs are embedding representations from a language model (Peters et al., 2018).
VanillaSpanBERT (SBERT): Joshi et. al (2019b) proposed a BERT (Devlin et al., 2018) version of the C2F model (Lee et al., 2018). Joshi et. al (2019b) introduced BERT to obtain all input embedding representations. For this baseline, the SpanBERT (Joshi et al., 2019a) variant of the model is used as the baseline owing to its performance gains.

4.2 Experimental Setup

The training set for these experiments is CEREC containing 6001 email threads and the validation set contains 34 email threads, the one used for coreference annotation. The SC containing 43 email threads is used as the test set. Mention detection and coreference resolution are the two tasks evaluated in these experiments. The following three experiments are carried out:

- Exp1: Use the Hb1 and Hb2 baselines for evaluating coreference resolution given mention annotations as input. Additionally, these baselines also use section information (SI) to identify mentions present in an email header.
- Exp2: Use the C2F and SBERT baselines to evaluate both mention detection and coreference resolution tasks. Compared to the SBERT baselines, the C2F baseline does not enforce a maximum sentence length restriction and has a higher hyperparameter value for maximum training sentences.

The *genre* feature is removed for both C2F and SBERT baselines since it does not apply to this corpus. For the C2F baseline, the hyperparameters *max_span_width*, *max_training_sentences* and epochs are set to 20, 30 and 10 respectively. This is done to make training tractable on the environment. For the SBERT baseline, the *spanbert_base* model is used with a maximum segment length of 256, and training is carried out on an NVIDIA GeForce GTX 1080 Ti GPU with 8 12gb cores.

4.3 Evaluation Metrics

This work follows the standard experimental setup used in the CoNLL 2012 Shared task. Primary evaluation is done using the unweighted average of MUC, B^3, and CEAFE metrics (Pradhan et al., 2012)[10]. In addition to this, scores using the LEA metric (Moosavi and Strube, 2016) are also reported.

4.4 Results

Table 6 shows results of Exp1 and Exp2 for all baselines. First, it can be seen that how first-person plural pronouns are resolved in the header baselines does not have a significant impact on the average F1 score. Second, the average F1 score of SBERT is just 0.23 F1 points higher than the C2F baseline. This shows that increasing the maximum sentence length and maximum training sentences do not help C2F in outperforming SBERT. Both models perform equally well. Compared to the results reported by Dakle et. al (2020), the SBERT baseline performs slightly better. Finally, the large difference in F1 scores of the Exp1 baselines and Exp2 baselines is because Exp1 baselines use mention annotations and the SI feature.

5 Error Analysis

This section presents error analysis performed on the predictions obtained by the baselines on a subset of 15 email threads selected randomly from SC. The selected 15 email threads contain a total of 282 coreference chains with 1261 mentions. To gain an in-depth understanding of the errors, human evaluation is performed. Errors are broadly divided into four categories. These are similar to the categories used by

[9] https://github.com/kentonl/e2e-coref
[10] https://github.com/conll/reference-coreference-scorers

	MUC			B^3			CEAFE			LEA			
Model	P	R	F1	P	R	F1	P	R	F1	P	R	F1	Avg. F1
Hb1	90.2	75.1	**81.9**	82	65.3	72.7	61.6	74.4	**67.4**	71.1	62	66.3	74
Hb2	91.3	74	81.8	87.1	64.2	**74**	59.2	76.6	66.8	75.7	60.3	**67.1**	**74.2**
C2F	82.9	68.7	75.2	52.5	53.5	**53**	65.2	22.5	33.5	49.1	52.8	**50.9**	53.9
SBERT	82.3	69.5	**75.3**	51.9	53.7	52.8	62.4	23.6	**34.2**	48.3	53.3	50.7	**54.1**

Table 6: Evaluation results on SC. Avg. F1 score is computed using MUC, B^3 and CEAFE metrics.

Aktaş et. al (2018) and Dakle et. al (2020) in their work. Table 7 shows the distribution of errors into these categories for each of the baselines.

5.1 Missing references in the chain

A reference that is present in a gold coreference chain but absent in the predicted chains is termed as a missing reference. Hb1 and Hb2 baselines use mention annotations as input to perform coreference chaining. Owing to this reason, only the deep learning baselines are considered for this error category. Missing references are further divided into three types to understand the limitations of the baselines.

1. Missing pronoun references: This error type contributes to 5-6% of all missing references.

2. Missing references in email header: A missing email address or name of a participant in the email message present in the email header is considered in this type. This error type contributes to 23-24% of all missing references.

3. Other missing references: All missing non-pronomial references present in the email body are considered in this error type. For C2F and SBERT, the distribution range of these missing references with respect to entity types is: PER - 23-30%, ORG - 19-23%, LOC - 16-19%, and DIG - 31-38%.

5.2 Missing chains

In this error category, coreference chains that are present in the gold annotations but absent in the predictions are considered. Since Hb1 and Hb2 use mention annotations as input, counts for this error category for these baselines are not reported. The models C2F and SBERT in the original work (Lee et al., 2018; Joshi et al., 2019b) were trained on CoNLL 2012 shared task corpus, which did not contain any singletons. Both C2F and SBERT baselines report similar numbers for this error category. About 82-85% of chains in this error category are of lengths 1 or 2.

5.3 Incorrectly chained references

All mentions in a coreference chain are considered to refer to the same entity. A mention or reference in a predicted coreference chain which does not refer to the same entity is considered to be incorrectly chained. These references are further broken down into pronoun references and other references. All baselines report a close count for pronoun references with C2F reporting the worst one. SpanBERT owing to its higher context capturing capabilities does a better job at resolving pronomial references than C2F. For other references, C2F and SBERT baselines report approximately 4 times the counts reported by Hb1 and Hb2. This highlights the effectiveness of rule-based approaches and the possible benefits of having a hybrid approach.

5.4 Decomposed chains

A gold coreference chain which is present in the predicted chains in the form of two or more chains is called as a decomposed chain. An email thread consists of multiple email messages. A model may perform well when the scope is restricted to a single email message but may fail to link entity chains belonging to different email messages. In addition to this, paraphrasing of a mention can also result

in multiple chains being created. Counts are reported for both the number of original chains and the number of chains that are created. It is evident by the high number of decomposed chains for Hb1 and Hb2 baselines that deep learning models do a better job of linking chains across email messages and handling paraphrasing. However, this also increases incorrectly chained references.

Moosavi et. al.(2016) in their work on coreference metrics, highlight the limitations of the CEAFE metric. Identifying entity mentions correctly but splitting a single chain into multiple parts can lower the CEAFE metric score. Compared to the rule-based systems, deep learning models identify less number of mentions with fewer chain splits (see Table 7). We recognize this as the reason for Exp2 models to obtain a higher CEAFE precision score over Exp1 systems. However, the large number of missing references significantly reduces the CEAFE recall for Exp2 models resulting in Exp1 models having a higher F1 score.

Error Category	Hb1	Hb2	C2F	SBERT
Missing references in the chain				
Missing pronoun references	-	-	15	13
Missing references in email header	-	-	60	59
Other missing references	-	-	170	159
Missing chains	-	-	80	81
Incorrectly chained references				
Pronouns	109	48	116	106
Other	83	78	309	312
Decomposed chains				
Number of chains decomposed	48	56	12	11
Number of new chains	117	148	25	24

Table 7: Error statistics of baselines for different error categories

6 Problem of missing context

Aktaş et. al (2018) and Dakle et. al (2020) highlight the challenges encountered for the entity resolution task in a conversational thread-like setting. This section points out an additional challenge corroborates on the difficulty of the task. Conversations using any media generally follow a tree-like structure, where multiple topics may branch off the initial topic but still follow a topic flow. In this flow, every message provides a piece of the whole context which helps in understanding the thread. The deletion of an intermediate message can result in creating ambiguity in the resolution of entities. The deletion of an intermediate email message not only results in the loss of the email text but also the loss of inclusion or exclusion of participants or change of email subject. Carenini et. al (2005) emphasized this issue in their work on the discovery of hidden emails.

7 Conclusion

This paper presents CEREC, the first large annotated corpus for the entity resolution in email conversations task. The corpus consists of 6001 email threads with 38,996 coreference chains. The two steps in the construction of the corpus along with the results of the experiments involved and statistics of the resulting corpus are explained. The construction process is carried out with minimal human intervention. We also evaluate the addition of features specific to text in a conversational thread-like setting. Two rule-based and two deep learning baselines are used for evaluation of the corpus. Qualitative and quantitative error analysis is presented on the predictions obtained using all baselines highlighting the avenues for improvement. Future work will consist of evaluating probable solutions for the entity resolution task. We also plan to conduct additional experiments to understand the effect of features presented in this paper using a larger corpus.

References

Berfin Aktaş, Tatjana Scheffler, and Manfred Stede. 2018. Anaphora resolution for twitter conversations: An exploratory study. In *Proceedings of the First Workshop on Computational Models of Reference, Anaphora and Coreference*, pages 1–10.

Sakhar Alkhereyf and Owen Rambow. 2017. Work hard, play hard: Email classification on the avocado and Enron corpora. In *Proceedings of TextGraphs-11: the Workshop on Graph-based Methods for Natural Language Processing*, pages 57–65, Vancouver, Canada, August. Association for Computational Linguistics.

Jake D Brutlag and Christopher Meek. 2000. Challenges of the email domain for text classification. In *ICML*, volume 2000, pages 103–110.

Giuseppe Carenini, Raymond T. Ng, and Xiaodong Zhou. 2005. Scalable discovery of hidden emails from large folders. In *Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining*, KDD '05, page 544–549, New York, NY, USA. Association for Computing Machinery.

Giuseppe Carenini, Raymond T Ng, and Xiaodong Zhou. 2007. Summarizing email conversations with clue words. In *Proceedings of the 16th international conference on World Wide Web*, pages 91–100.

Yu-Hsin Chen and Jinho D Choi. 2016. Character identification on multiparty conversation: Identifying mentions of characters in tv shows. In *Proceedings of the 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 90–100.

Kevin Clark and Christopher D. Manning. 2015. Entity-centric coreference resolution with model stacking. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1405–1415, Beijing, China, July. Association for Computational Linguistics.

William W Cohen et al. 1996. Learning rules that classify e-mail. In *AAAI spring symposium on machine learning in information access*, volume 18, page 25. Stanford, CA.

William W Cohen, Vitor R Carvalho, and Tom M Mitchell. 2004. Learning to classify email into "speech acts". In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 309–316.

Simon Corston-Oliver, Eric Ringger, Michael Gamon, and Richard Campbell. 2004. Task-focused summarization of email. In *Text Summarization Branches Out*, pages 43–50.

Agata Cybulska and Piek Vossen. 2014. Guidelines for ecb+ annotation of events and their coreference. Technical report, Technical Report NWR-2014-1, VU University Amsterdam.

Parag Pravin Dakle, Takshak Desai, and Dan Moldovan. 2020. A study on entity resolution for email conversations. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 65–73, Marseille, France, May. European Language Resources Association.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

David Kirsch Sergey Golitsynskiy Douglas Oard, William Webber. 2015. Avocado research email collection. *Philadelphia: Linguistic Data Consortium*.

Ralph Grishman and Beth Sundheim. 1996. Message understanding conference- 6: A brief history. In *COLING 1996 Volume 1: The 16th International Conference on Computational Linguistics*.

Iris Hendrickx and Veronique Hoste. 2009. Coreference resolution on blogs and commented news. In *Discourse Anaphora and Anaphor Resolution Colloquium*, pages 43–53. Springer.

Yifen Huang and Tom M Mitchell. 2008. Exploring hierarchical user feedback in email clustering. In *Enhanced Messaging Workshop in Proceedings of the 23rd AAAI Conference on Artificial Intelligence (AAAI 2008)*.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S Weld, Luke Zettlemoyer, and Omer Levy. 2019a. Spanbert: Improving pre-training by representing and predicting spans. *arXiv preprint arXiv:1907.10529*.

Mandar Joshi, Omer Levy, Daniel S Weld, and Luke Zettlemoyer. 2019b. Bert for coreference resolution: Baselines and analysis. *arXiv preprint arXiv:1908.09091*.

Bryan Klimt and Yiming Yang. 2004. The enron corpus: A new dataset for email classification research. In *European Conference on Machine Learning*, pages 217–226. Springer.

Derek Scott Lam. 2002. *Exploiting e-mail structure to improve summarization*. Ph.D. thesis, Massachusetts Institute of Technology.

Kenton Lee, Luheng He, and Luke Zettlemoyer. 2018. Higher-order coreference resolution with coarse-to-fine inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 687–692, New Orleans, Louisiana, June. Association for Computational Linguistics.

David D Lewis and Kimberly A Knowles. 1997. Threading electronic mail: A preliminary study. *Information processing & management*, 33(2):209–217.

Giuseppe Manco, Elio Masciari, Massimo Ruffolo, and Andrea Tagarelli. 2002. Towards an adaptive mail classifier. In *Proc. of Italian Association for Artificial Intelligence Workshop*.

Einat Minkov, Ramnath Balasubramanyan, William W Cohen, and Machine Learning Dep. 2008. Activity-centric search in email. In *Enhanced Messaging Workshop, AAAI*.

Nafise Sadat Moosavi and Michael Strube. 2016. Which coreference evaluation metric do you trust? a proposal for a link-based entity aware metric. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 632–642, Berlin, Germany, August. Association for Computational Linguistics.

Hiroyuki Murakoshi, Akira Shimazu, and Koichiro Ochimizu. 2000. Construction of deliberation structure in e-mail communication. *Computational Intelligence*, 16(4):570–577.

Smaranda Muresan, Evelyne Tzoukermann, and Judith L. Klavans. 2001. Combining linguistic and machine learning techniques for email summarization. In *Proceedings of the ACL 2001 Workshop on Computational Natural Language Learning (ConLL)*.

Ani Nenkova and Amit Bagga. 2004. Facilitating email thread access by extractive summary generation. *Recent advances in natural language processing III: selected papers from RANLP*, 2003:287–294.

Paula S Newman and John C Blitzer. 2003. Summarizing archived discussions: a beginning. In *Proceedings of the 8th international conference on Intelligent user interfaces*, pages 273–276.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana, June. Association for Computational Linguistics.

Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Olga Uryupina, and Yuchen Zhang. 2012. CoNLL-2012 shared task: Modeling multilingual unrestricted coreference in OntoNotes. In *Joint Conference on EMNLP and CoNLL - Shared Task*, pages 1–40, Jeju Island, Korea, July. Association for Computational Linguistics.

Owen Rambow, Lokesh Shrestha, John Chen, and Chirsty Lauridsen. 2004. Summarizing email threads. In *Proceedings of HLT-NAACL 2004: Short Papers*, HLT-NAACL-Short '04, page 105–108, USA. Association for Computational Linguistics.

Marta Recasens, Lluís Màrquez, Emili Sapena, M Antònia Martí, Mariona Taulé, Véronique Hoste, Massimo Poesio, and Yannick Versley. 2010. Semeval-2010 task 1: Coreference resolution in multiple languages. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 1–8. Association for Computational Linguistics.

Ian Soboroff, Arjen P de Vries, and Nick Craswell. 2006. Overview of the trec 2006 enterprise track. In *Trec*, volume 6, pages 1–20.

Jan Ulrich, Gabriel Murray, and Giuseppe Carenini. 2008. A publicly available annotated corpus for supervised email summarization. In *Proc. of aaai email-2008 workshop, chicago, usa*.

Steve Whittaker and Candace Sidner. 1996. Email overload: exploring personal information management of email. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 276–283.