electronics

MDPI

Article

Reply Using Past Replies—A Deep Learning-Based E-Mail Client

Yiwei Feng [2], M. Asif Naeem [1,2,*], Farhaan Mirza [2] and Ali Tahir [3]

[1] Department of Computer Science, National University of Computer and Emerging Sciences, Islamabad 44000, Pakistan
[2] School of Engineering, Computer & Mathematical Sciences, Auckland University of Technology, Auckland 1142, New Zealand; fxyfeier@gmail.com (Y.F.); farhaan.mirza@aut.ac.nz (F.M.)
[3] Institute of Geographical Information Systems (IGIS), National University of Sciences and Technology (NUST), Islamabad 24090, Pakistan; ali.tahir@igis.nust.edu.pk
* Correspondence: asif.naeem@nu.edu.pk or mnaeem@aut.ac.nz

Received: 2 July 2020; Accepted: 18 August 2020; Published: 20 August 2020

Abstract: Email is the most common and effective source of communication for most enterprises and individuals. In the corporate sector the volume of email received daily is significant while timely reply of each email is important. This generates a huge amount of work for the organisation, in particular for the staff located in the help-desk role. In this paper we present a novel Smart E-mail Management System (SEMS) for handling the issue of E-mail overload. The Term Frequency-Inverse Document Frequency (TF-IDF) model was used for designing a Smart Email Client in previous research. Since TF-IDF does not consider semantics between words, the replies suggested by the model are not very accurate. In this paper we apply Document to Vector (Doc2Vec) and introduce a novel Gated Recurrent Unit Sentence to Vector (GRU-Sent2Vec), which is a hybrid model by combining GRU and Sent2Vec. Both models are more intelligent as compared to TF-IDF. We compare our results from both models with TF-IDF. The Doc2Vec model performs the best on predicting a response for a similar new incoming Email. In our case, since the dataset is too small to require a deep learning algorithm model, the GRU-Sent2Vec hybrid model cannot produce ideal results, whereas in our understanding it is a robust method for long-text prediction.

Keywords: email management system; deep learning; word embedding

1. Introduction

E-mail is still the most common form of online business correspondence and is still a growing and effective communication tool for most enterprises and individuals [1]. Meanwhile, E-mail is also an integral part of related personal Internet experience [2]. For example, E-mail accounts (or E-mail addresses) are almost always required for registering on website accounts, including social networking sites, instant messaging and any other types of Internet services. Therefore, E-mail has become fully integrated into our daily lives and business activities. According to the Radicati Group's statistics and projections [1], more than 281 billion E-mails are sent and received worldwide every day, and this number is expected to increase by 18.5 per cent over the next four years. In 2018, more than half of the world's population used E-mail, with more than 3.8 billion users. Based on the above statement, it can be determined that an average user sends and receives an average of 74 E-mails per day, which also reveals the problem of E-mail overload.

The problem of E-mail overload has not been solved for nearly half a century [3]. Most users have numerous E-mails that they do not ever read or receive a reply to in time, which leads to E-mail management issues, predominantly a messy and overwhelming mailbox [3]. Enterprises still maintain

and manage customer resources using E-mails because of handling users' feedback and consultation [2]. To be specific, some customer service centres of various organisations receive hundreds of thousands of E-mails from customers every day. Although the staff in the customer service centres have high-level training, there is striking similarity among huge E-mail data. They need to spend lots of time replying to these E-mails, which results in a high labour cost during this process for enterprises. Every E-mail user also suffers from E-mail overload. The development of the information age brings various benefits to our daily life, however, alongside its convenience, attendant problems are equally persistent.

For E-mail systems, E-mail overload was proclaimed as a 'universal problem' [3]. As a formal means of communication, E-mail is 'central' [4], 'ubiquitous' [5] and 'indispensable' [6]. Whittaker and Sidner [3] presented that E-mail users tend to leave an increasing volume of unread or non-replied messages every day. Most users have numerous E-mails that they do not ever read or get to reply to in time, which leads to E-mail management issues, predominantly a messy and overwhelming mailbox.

Researchers believe that NLP techniques using machine learning and deep learning algorithms have a significant role in reducing time and labour wasted due to repeated E-mail responses. They focus on developing E-mail systems with intelligent response functions. However, there are still research gaps either in reusing old E-mails based on an information retrieval method or the research of predictive generation-based responses based on neural networks. Even Gmail, now one of the best at responding intelligently to E-mails, uses technology based on word-level predictions. However, most E-mails are long-text; even the shortest E-mails usually are more than two sentences long. The research in this area still lacks ability to predict and generate E-mail reply consisting multiple sentences. Therefore this is our motivation of our research to postulate a novel EMS.

Our research objective is to present an intelligent E-mail response solution for individuals or corporate departments (such as service centres or help desks) that receive a large number of similar E-mails every day. This study contributes to the improvement of the existing models towards relevant practical usage scenarios. The analysis of this study identified a group of three essential models that can be applied in reducing E-mail overload. The TF-IDF model was applied in E-mail systems in 2017 [7], there are few limitations such as not marking E-mail labels and too much noise in the datasets. Holistic analysis of the benefits of the Doc2Vec model has not been done before. Certainly, for the GRU-Sent2Vec hybrid model, it uses a combination of information generation and retrieval, which is an innovative model and is proposed for the first time. Therefore, the study confirmed our results of innovative research that also emphasised the contributions of three aspects. The paper made the following key contributions:

1. Current popular methods are limited to short text prediction, at the word-level. For example *Chatbots* can predict the next sentence based on the last sentence. However, E-mails are long-text, and if the content of the reply can be predicted from the received E-mail, work efficiency of E-mail users will be improved in particularly considering the customer services in both public and private corporate sectors. In this paper, Sent2Vec is combined with GRU and a novel hybrid model is constructed to make predictions based at the sentence-level rather than word-level. However, it should be noted that the scenarios applied by the models in this research are not limited to long text prediction but are also suited for short-text prediction scenarios, such as Chatbots used in automatic reply in a chat room.
2. Using mapping rules to improve the training corpus. The E-mail dataset is processed using a logical matching method, which matches sent and received E-mails and uses them as a reference for answering new E-mails. This study expands the sample diversity of the public E-mail dataset, and the processed dataset will be published on GitHub https://github.com/fxyfeier for further research.
3. By creating user interfaces to implement core functions, our research is truly applicable for many enterprise customer service departments. This intelligent E-mail reply suggestion system allows users to choose an intelligent reply function or direct reply. The system provides an opportunity for users to review automatically generated replies before they are sent.

4. We evaluate the performance of Doc2Vec and GRU-Sent2Vec with TF-IDF. We also carry out human-based evaluation to validate our results.

The rest of the paper is structure as follows. Section 2 presents literature review about the relevant approaches. Section 3 describes design and methodology for our proposed approach. Section 4 presents implementation and evaluation of our SEMS. Finally Section 5 concludes the paper.

2. Related Work

2.1. New E-Mail Response Approaches

Automatic reply generation is a tedious and frustrating process, past researchers explored three areas including predicting response behaviour, reusing Previous Reply E-mails, and automatically generating E-mail Response.

2.1.1. Predicting Response Behaviour

Dredze et al. [8] developed a prototype of an E-mail system that could predict the response behaviour to an E-mail. Yang et al. [9] found various features that affect E-mail response behaviour, such as E-mail content, meta-data factors, time series features. The primary method of this predictive response behaviour model is to classify E-mails based on their subjects or extracted content. Feature extraction is done by performing tag checking. First, the question keywords in the E-mail content, the question marks, the E-mail addresses, some particular keywords (such as attach, attachments, attached) in the E-mail are marked as special identifying information [8,10]. Most of these models adopt rule-based classification technology. However, the necessary condition of classification optimisation is a large amount of labelled training data, so it is an essential and challenging task to collect or process enough training datasets to construct a classifier knowledge base. Therefore, based on the limitations of the training corpus, we have excluded this approach in our research.

2.1.2. Reusing Previous Reply E-Mails

The second primary approach which is also known as textual Case-Based Reasoning (CBR) was designed to reuse previous reply E-mails [11,12]. The method is inspired by the scientific study of human memory cognition [13], using previous experience to solve similar new problems. Lapalme and Kosseim [11] divided CBR processing into three stages, namely case retrieval, case reuse and answer penalisation. Furthermore collected all basic word pairs from the received E-mails and their replies to selected the most crucial word pairs. In this process, entities such as the sender name, company name, and specific business need to be tagged with the corresponding reply content using senders, subsidiaries, or financial institutions provided in the prepared repository [11]. Hewlett and Freed [14] used the Margin Infused Relaxed Algorithm (MIRA) to deal with multi-class problems. It is a good design idea to reuse a similar previous E-mail as a new response. Although there is not much research that can be used for a reference regarding this method, this research will draw on the ideas of previous studies.

2.1.3. Automatically Generating E-Mail Response

In recent years, artificial intelligence has made rapid progress in Natural Language Processing (NLP). The Google team [15] present a new deep learning algorithm, namely, the Recurrent Neural Network (RNN) [16], was implemented for the design of the auto-reply E-mail function. They used the Long Short-Term Memory (LSTM) neural network (an improved model of the RNN) [17] to process received messages and predict responses. Considering the high training cost of the LSTM neural network model. In combination with machine learning classification, weighted keywords and similarity measurement techniques, Parameswaran et al. [18] designed the function of automatically generating and suggesting short E-mails. In May 2018, Google's research team improved their

auto-reply model on Gmail's smart prediction function https://ai.googleblog.com/2018/05/smart-compose-using-neural-networks-to.html. This impressive function has dramatically improved the Gmail user experience. It can predict the next word in a sentence that a user might want to type, based on the order of the preceding words. The core technology they used is still the LSTM neural network, which combines the natural Bag-of-Words (BoW) [19] method to balance delay constraints.

It is challenging approach because deep neural networks have higher requirements in all aspects, such as the quality of training data. If the dataset is not large enough, then this high-level neural network may not learn relevance properly. Meanwhile, since it is an end-to-end unsupervised learning method, we cannot control the learning process and results. Nevertheless, although the dataset we can use is small, we still want to try the most advanced approach in this study. In order to make up the deficiency of objective conditions, we will introduce some new ideas to optimise the model structure.

2.2. Related Techniques

In reviewing relevant techniques from previous studies, we identified two main approaches to designing our models, information retrieval and information generation. For the retrieval-based model, this project adopted TF-IDF and Doc2Vec algorithms for experiments.

TF-IDF is a popular term weighting scheme based on the Bag-of-words statistics widely used in information retrieval and text mining [20,21]. Assuming the document is just a collection of words, the document can be vectorised by calculating the $TF - IDF_{(n,d)}$ value of each word. The vectorised document refers to the vector space model, which allows calculation of the similarity between all documents in a corpus by using the Cosine theorem. In a very recent paper, Kim et al. proposed a multi-co-training (MCT) approach in the field of document classification [22], combining TF-IDF, latent Dirichlet allocation (LDA) and Doc2Vec based on shallow neural networks. The combination of these three methods increased the diversity of feature sets used for classification. They also analysed and compared the characteristics of each method in their research.

Doc2Vec or Sent2Vec is an extension of Word2Vec [23], similar to Word2Vec, except that it uses a fixed-length vector to represent an entire document or a sentence. This unsupervised learning algorithm can express paragraphs or documents as vectors that are well suited for document processing tasks. For example, it can be used to compare similarities between paragraphs or documents. There are four modes in the Doc2Vec model. Distributed Memory Model of Paragraph Vectors (PV-DM) architecture, Paragraph Vector-Distributed Bag of Words (PV-DBOW) structure, Hierarchical Softmax (HS) and Negative Sampling (NS) method [24]. Due to its good performance, Doc2Vec has become more and more popular in the industry in recent years.

In terms of the second approach mentioned above, we selected a deep learning algorithm to design our generative model. With the rapid improvement of hardware that supports extensive capabilities and advances in deep learning technology, many researchers proposed various novel neural network algorithms. Especially after the concept of distributed representations of word vector [24] was widely accepted, Mikolov et al. [25] found that many results have verified those language models trained by using neural network based on big datasets are significantly superior to traditional language models in terms of performance. They also stated that deep learning methods had achieved very high performance across different NLP tasks. Since the sequence-to-sequence (seq2seq) model was successfully introduced from the field of machine translation into the Chatbot dialogue system [26], we tried to apply a seq2seq model, GRU, to our Smart E-mail Management System.

Chung et al. [27] proposed the structure of the GRU, which was designed as a reset gate that it is enable to data learning and updating. It is much easier to implement than LSTM because GRU removes the cell state, then uses a hidden state to transfer information, and relatively reduces the amount of parameter tuning tasks during data modelling and training tasks [27]. For most E-mails are long text, research and exploration in this area are still lacking regarding how to predict and generate an E-mail consisting of multiple sentences.

In order to obtain better experimental performance, we improved some algorithm aspects, such as nested Sent2Vec into GRU, and designed a hybrid model of information generation combined with information retrieval.

3. Design and Methodology

The system framework of this experiment is shown in Figure 1. Initially, we needed to execute a series of text prepossessing steps on the original E-mail dataset. For the processed E-mails, we only kept the matching E-mail pairs (namely received E-mail–reply E-mail), so that we could obtain the relationships between the received E-mails and reply E-mails and put them into two databases. The next step is to train our three models using the results of text processing. Upon completion of the model training, when a new E-mail is received, SEMS client will offer users two options: one is to reply directly, and the other is to use the smart reply function. Regardless of whether users directly reply to an E-mail or modify the response suggestions generated by the models, after replying, the *newly received E-mail* will be paired with the *sent E-mail*. Finally the results will be stored in both databases in preparation for the models' future learning.

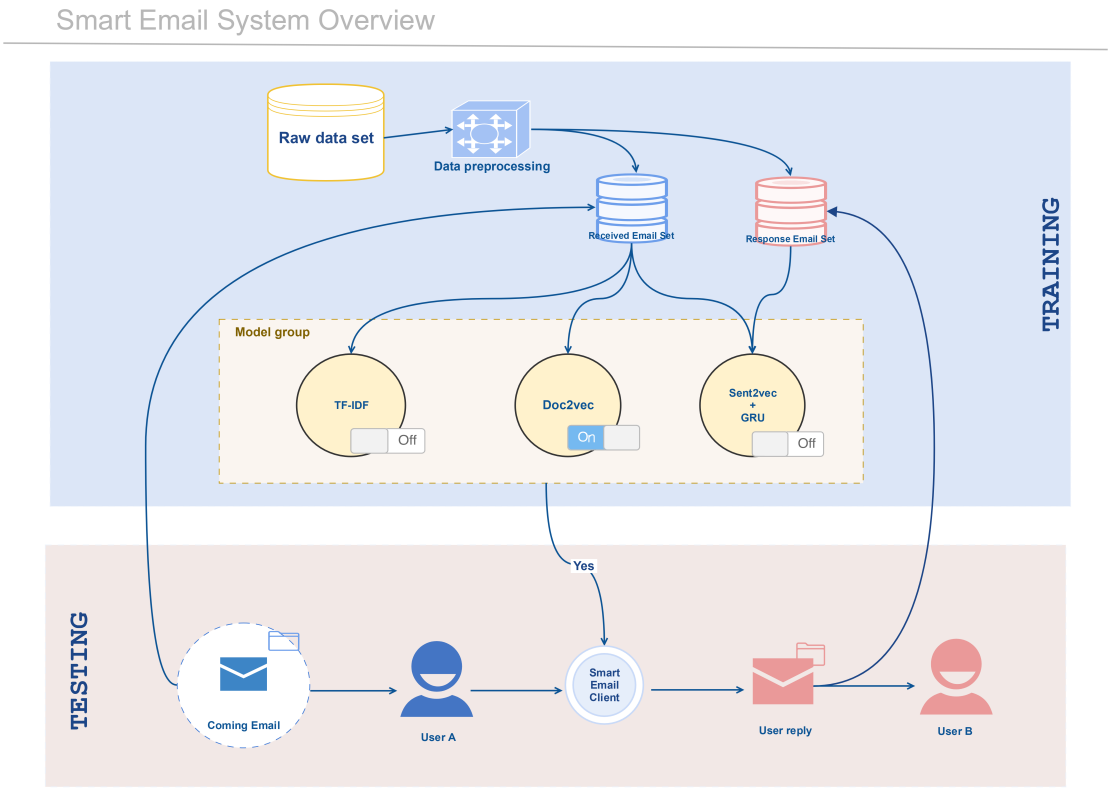

Figure 1. Smart Email Management System (SEMS) Overview.

3.1. TF-IDF Based Model

Figure 2 presents the workflow design for TF-IDF modelling. Firstly, after general dataset processing (which described in the next section), *Received E-mail Dataset* and *Response E-mail Dataset* are generated. Based on these datasets the data is not needed to be processed further to adapt to the different models. Data Processing for TF-IDF includes word lower-casing, word tokenization, stop words removal and word stemming. In the modelling process, vocabulary-building is a way to tag a collection of text documents, index each known word, and encode new documents using the index set as well.

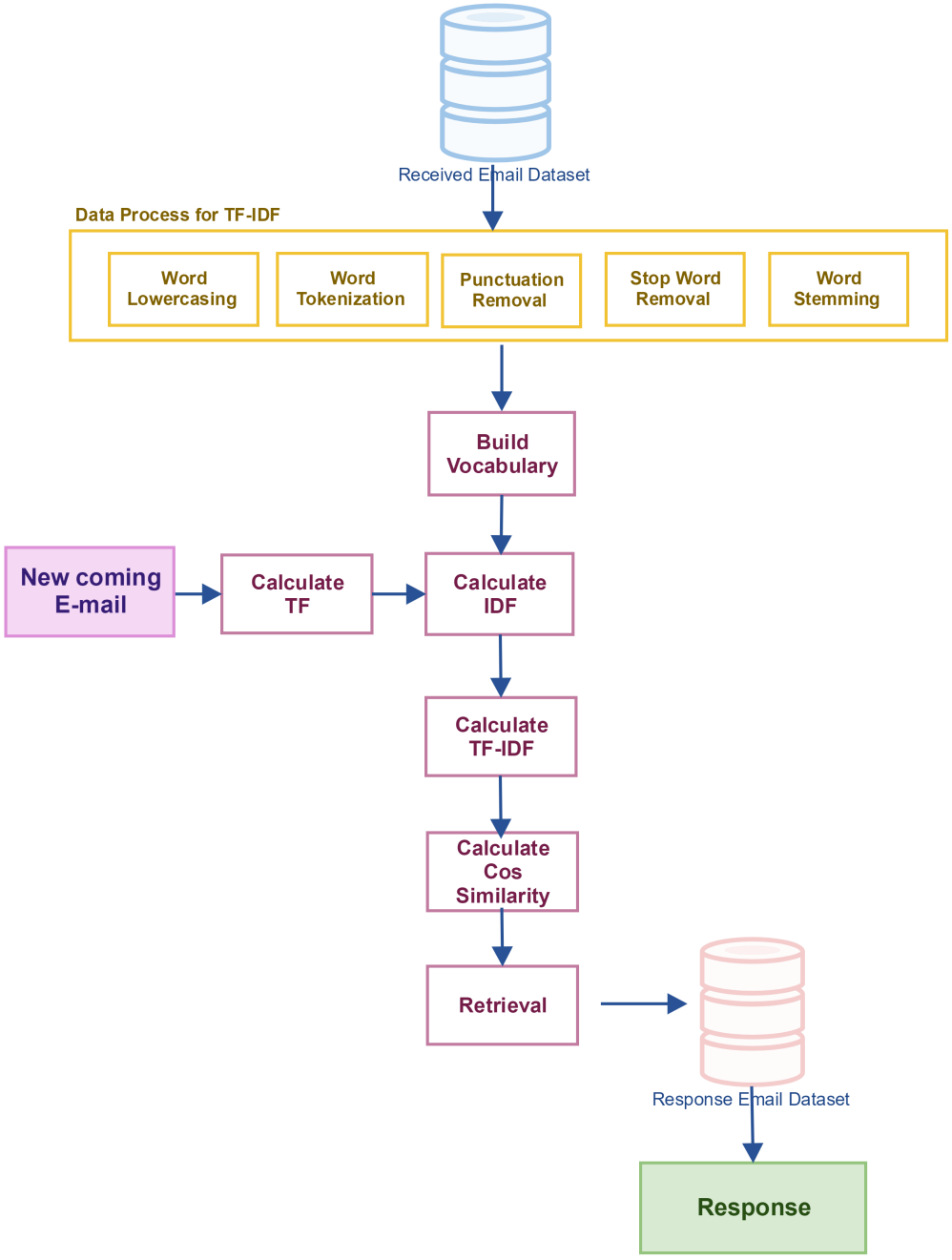

Figure 2. Term Frequency-Inverse Document Frequency (TF-IDF) Based Model.

After building corpus vocabulary and a sparse matrix, the next step is to count the IDF value and TF-IDF value of each word in the given corpus as well as a new document.For a new document, it is only needed to calculate its TF-IDF vector related to the entire corpus. By dot product with all other document vectors, we could calculate the cosine distance between a new document and all documents in the corpus, so that we could sort similar documents for retrieval.

Calculating IDF and TF-IDF Scores: After building corpus vocabulary and a sparse matrix by CountVectorizer, TF-IDFTransformer was used to count the IDF value and TF-IDF value of each word in the given corpus as well as a new document. It should be noted that if some specific words do not appear in the training corpus, their TF-IDF values can be 0.

Calculating Similarity Scores: Euclidean Normalisation is also applied in the Scikit-learn library. For a new document, we only needed to calculate its TF-IDF vector related to the entire corpus. By dot product with all other document vectors, we could calculate the cosine distance between a new document and all documents in the corpus, so that we could sort similar documents for retrieval.

3.2. Doc2Vec Based Model

Figure 3 interprets the training workflow based on the Doc2Vec model. As for data processing, it is similar to the process of TF-IDF.

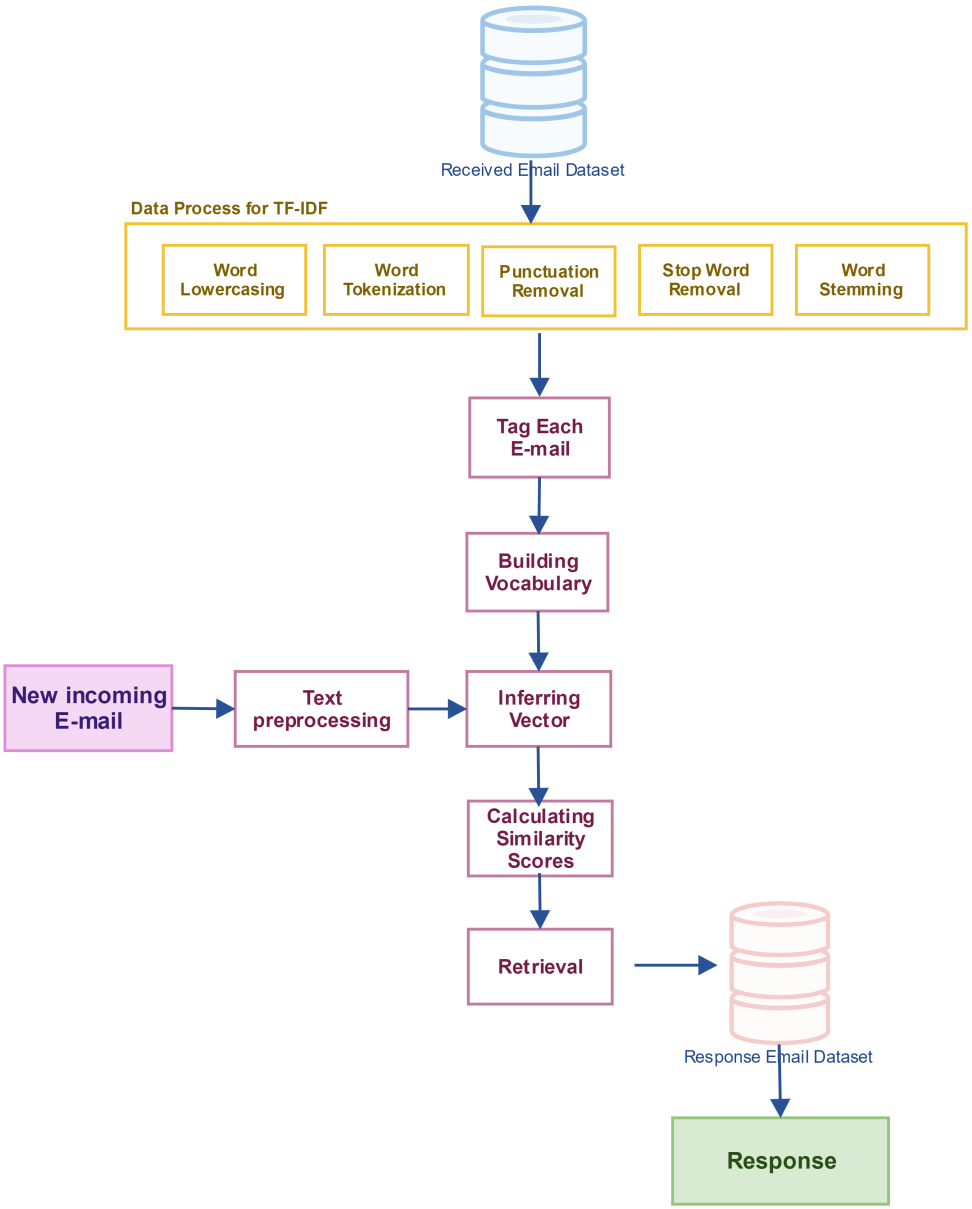

Figure 3. Doc2Vec Based Model.

Before training the model, we needed to tag each document (E-mail) in the corpus. To get the correlations between words in the document, we embedded the document's ID with the words in the document. After building vocabulary, inferring vectors is the way to transfer the new documents to vectors. Using them makes it easy to calculate similarity scores among documents.

Tag Each E-mail: Before training the model, we needed to tag each document (E-mail) in the corpus. To get the correlations between words in the document, we embedded the document's ID with the words in the document. Building Vocabulary: In the Gensim library, the function Word2VecVocab can create a vocabulary for the model. In addition to recording all unique words, this object provides additional functionality such as creating a Huffman tree (the more frequent the words, the closer they are to the roots of the tree), to eliminate uncommon words.

Inferring Vector: After training the model, the infer_vector function can infer vectors for new documents.

Calculating Similarity Scores: A built-in most_similar module in Gensim is used to calculate the similarity of document vectors.

3.3. GRU-Sent2Vec Hybrid Model

The essential part of GRU-Sent2Vec hybrid model is to put pre-trained sentence vectors into the embedding layer of GRU network for training together. The design idea of this model is to combine the information generation model with the information retrieval model, using both GRU and Sent2Vec technologies. In this design (Figure 4), on the one hand, we expect to generate long-text reply E-mails, on the other side, we consider that our training data set is too small for the deep neural network. It is a novel attempt for intelligent generated E-mail system. The model of Sent2Vec is built to train the vector of each sentence. The process and principles of training are similar to Doc2Vec. Whereas, the difference is that the ID of the document is replaced with the ID of the sentence for training. We mapped each sentence into a 150-dimensional feature space and added an attention layer to allow the decoder to focus on certain parts of the input sequence to improve the decoder's capability and prevent the loss of valuable information. In order to achieve better convergence, we used the teacher forcing method in iterative training.

Different from the previous two models, instead of a simple normalisation process, we continued to adopt Sent2Vec model to match similar sentences in the sentence list of the original training dataset. In the case of a small sample set, the intention is to ensure that the results generated by the GRU model are always valid.

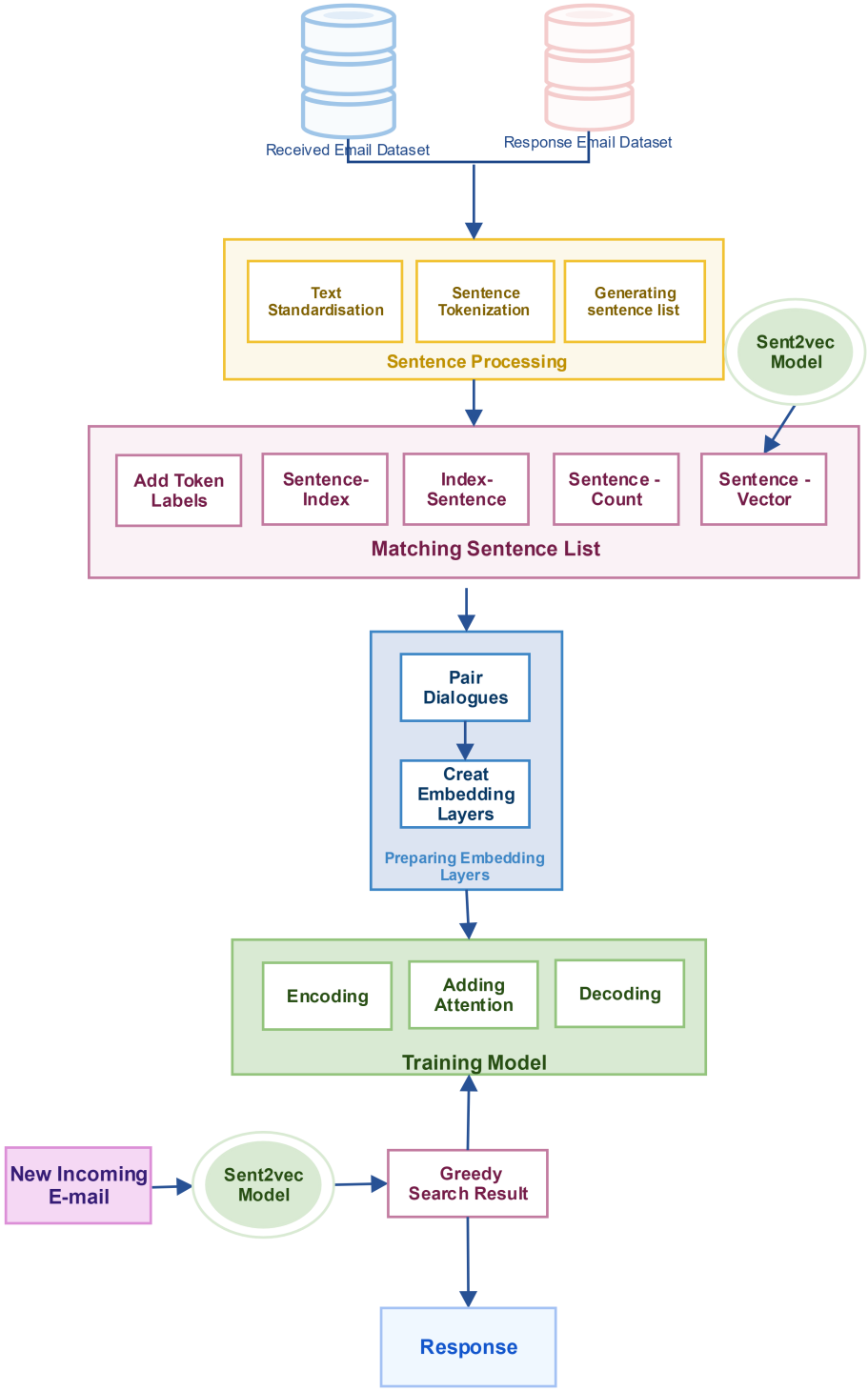

Figure 4. Gated Recurrent Unit Sentence to Vector (GRU-Sent2Vec) Hybrid Model.

4. Implementation and Evaluation

4.1. Data Preparing

Figure 5 shows the workflow of the data pre-processing. The purpose of this process is to produce a generic dataset for our three models.

The Enron E-mail Dataset https://www.cs.cmu.edu/~enron/ is a suitable training dataset available online. This dataset is composed of a large number of Emails in TXT form. The first step is to transfer all the E-mails into a CSV file with the information we needed, including the sending date, sender E-mail address, receiver E-mail address, E-mail subject and E-mail content, respectively.

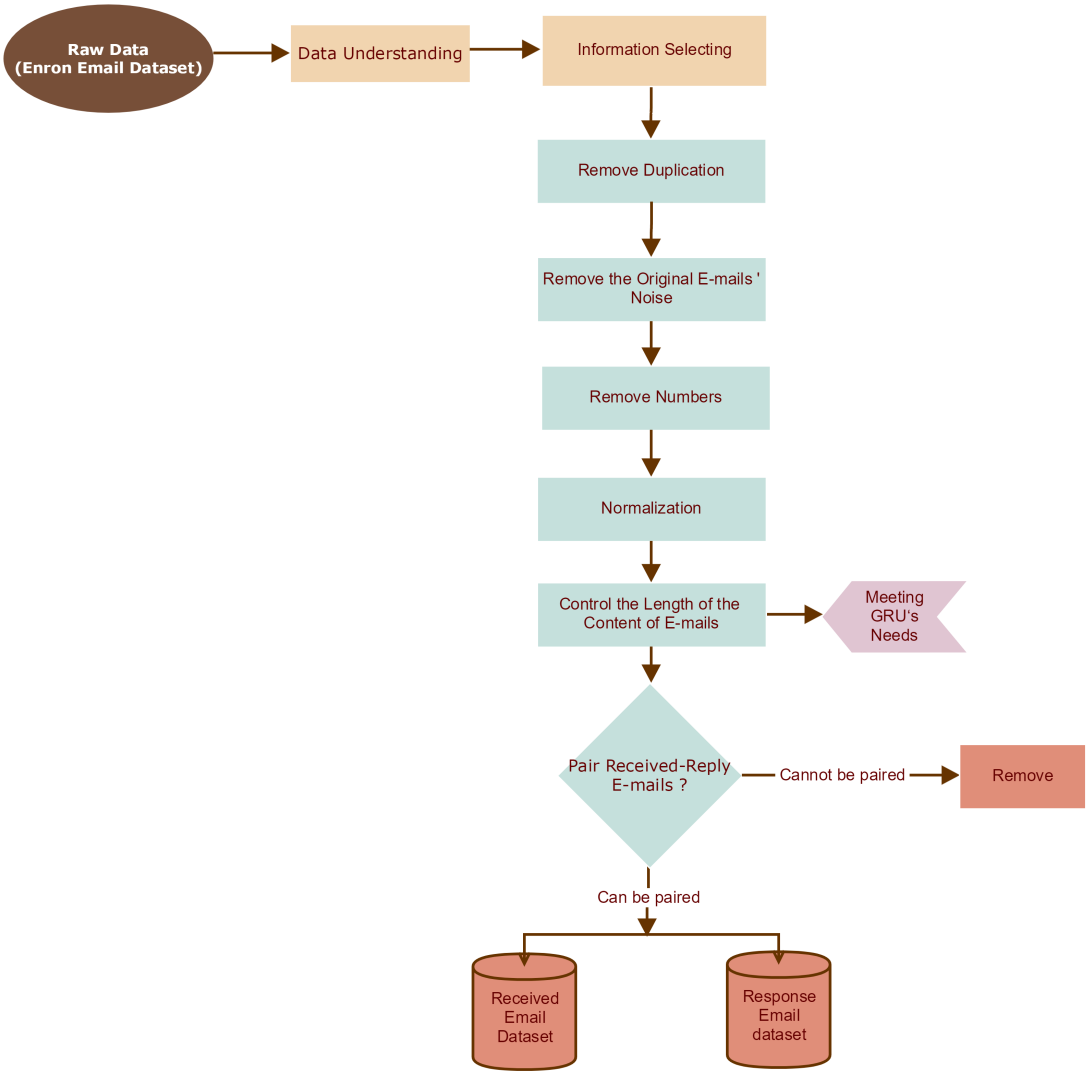

Figure 5. Data processing flowchart.

From the data overview (Figure 6), it can be seen that the total number rows in the dataset is 517,401, among which there are 498,214 messages with titles, and the highest frequency of occurrence is 'RE:', appearing 6477 times. Meanwhile, in the content column of the E-mail dataset, nearly half of the content is duplicate. The reason is that if employee A sends an E-mail to employee B, the E-mail in A's outbox will be the same as the E-mail received in B's inbox. Therefore, duplicate messages firstly should be removed entirely, leaving unique messages for further processing. The process of this step is to compare the title and content at the same time to avoid the possibility that the content is the same but not sent by the same person.

	Date	From	To	Subject	Content
count	517401	517401	495554	498214	516245
unique	224122	20328	55385	159289	241859
top	2001-06-27 23:02:00	frozenset({'kay.mann@enron.com'})	frozenset({'pete.davis@enron.com'})	RE:	The request has been completed with all resour...
freq	1118	16735	9155	6477	148

Figure 6. Data understanding.

The next step is to remove noise from the dataset, such as numbers, symbols, and the duplicate past responses in Emails because such non-useful information will lead to poor model training results.

The last but vital step is Pairing Received - Response E-mails, as it is the basic form for three models training. The method used in the E-mail pairing process is that we logically filtered messages using the sending time, title, and the name of the senders and receivers, then selected the E-mails that most likely relate to each other and placed them into two databases (Received E-mail Dataset and Response E-mail Dataset) separately. The result after processing is shown in Figure 7. Finally, we got a total of 19,871 pairs of E-mails, each with a content-length range of 1 to 30 sentences (as shown in Figure 8).

After general data processing, we need to further process the data to fit different models. The data processing methods for TF-IDF and Doc2Vec model are similar, including word lowercasing, word tokenization, stop word removal, word stemming. However, processing data for GRU models requires more effort. We need to separate each sentence in order to generate a sentence dictionary that is related to the unique index, and then apply the Sent2Vec model to train the sentence vectors, which will be used as an embedding layer of the GRU model. Therefore, the process of processing includes text standardisation, sentence tokenisation, generating sentence list, building up Sent2Vec model, mapping sentences and preparing embedding layers.

	Received	Response
0	Randy Can you send me a schedule of the salary...	Phillip I m working on getting the official li...
1	resumes of whom ?	The commercial support people that you and Hun...
2	Christy I read these points and they definitel...	Phillip To the extent that we can give Chair H...
3	Phillip I have a meeting tomorrow morning with...	Yes you can use this chart . Does it make sense ?
4	Phillip Which one should I do the x is half th...	I like the cedar t version better . Why don t ...
5	Phillip The Social Security and Medicare tax h...	Thanks .
6	Phillip This is the candidate I spoke with you...	Adrianne I cannot download his resume . Please...
7	Phillip Lets try this one . .Thanks ! Adrianne .	Left message and sent email . I will let you k...
8	Phillip I need to get the contract for Galaxy ...	Greg I would rather sign and fax copies of the...
9	Phillip I am waiting to get info . on two more...	Jeff Can you resend the info on the three prop...
10	Bernie Good Morning . I hope all is well . I h...	Kirk I ve added my comments . See attached Giv...

Figure 7. Received-Response E-mail Pairs.

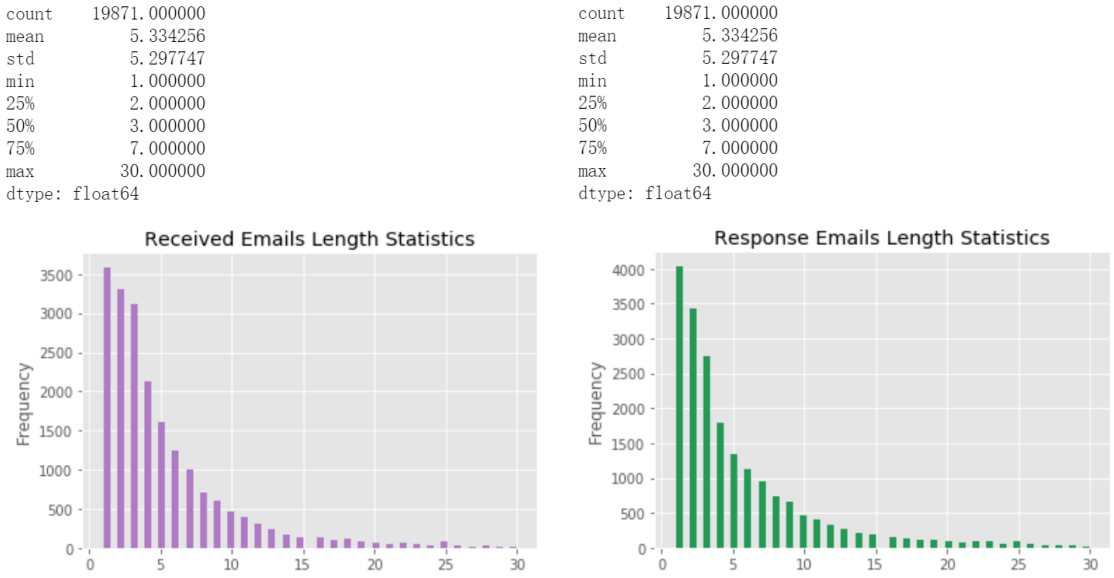

Figure 8. Processed E-mail Content Length Statistics.

4.2. Parameter Tuning

According to the design flow of the three models in the previous section, we trained the models respectively, and use the method of call@K to self-evaluate the models. Here we only perform parameter tuning of TF-IDF and Doc2Vec models. We do not consider GRU-Sent2Vec model because of the limitations, which will be discussed later.

This process randomly selects 200 documents from the training corpus as queries, and then carries out vector inference on these documents and compared them with the vectors in the training corpus. This self-evaluation process is based on the similarity level between the same documents and the query.

We assume that the test dataset consisting of these 200 documents is some new data, and then evaluate them based on the models' response to them. The expected result is that the same document in the training set will be extracted in either the first or the first three positions for the 200 test queries. The formula is expressed as Equation (1):

$$\begin{aligned} \text{Recall rate@1} &= \frac{\text{Retrieved documents in top 1}}{200} \\ \text{Recall rate@3} &= \frac{\text{Retrieved documents in top 3}}{200} \end{aligned} \tag{1}$$

We find that the most influential parameter for this model is N-gram size (Table 1). The higher the value of N, the higher the accuracy (Recall rate@1). From the results, it seems not to have much impact on the recall rate of the top three. However, in the meantime, the training time has an exponential increase. Considering the range of our training dataset is not large, we ignore the time consumption and selected Ngram = (1, 4) as the model parameter of TF-IDF. The results are shown in Table 1.

Table 1. Comparison of TF-IDF Parameters.

TF-IDF	Ngram	Training Time	Retrieved Numbers			Recall Rate @1	Recall Rate @3
			0	1	2		
1	(1,1)	1.8910167 s	173	12	5	86.5%	95%
2	(1,2)	5.3181312 s	173	11	4	86.5%	94%
3	(1,3)	10.235551 s	174	12	3	87%	94.5%
4	(1,4)	15.074302 s	175	10	5	87.5%	95%

Compared with TF-IDF, the Doc2Vec model contains more parameters. As for these parameters, eleven combinations were selected to conduct eleven rounds of training for the model (M1–M11). The results of the test are illustrated in Table 2 and Figure 9.

Table 2. Comparison of Doc2Vec Parameters (The best results are bold).

NO.	Doc2Vec Model	Vector_Size	Windows	Epochs	Training Time	Retrieved Numbers			Recall @1	Recall @3
						0	1	2		
1	NS + DM	100	5	500	346.51621 s	174	5	3	87%	91%
2	NS + DM	100	5	1000	698.35794 s	173	9	5	86.5%	93.5%
3	**NS + DM**	100	5	1500	1013.3812 s	173	11	5	86.5%	**94.5%**
4	NS + DM	150	5	1500	1059.6635 s	175	8	4	87.5%	93.5%
5	NS + DM	200	5	1500	1019.1251 s	176	7	2	88%	92.5%
6	NS + DM	150	10	1500	1042.714 s	175	9	3	87.5%	93.5%
7	NS + DM	150	5	2000	1407.6521 s	172	12	2	86%	93%
8	HS +DM	150	5	1500	1196.5312 s	171	9	6	85.5%	93%
9	**NS + DBOW**	150	5	1500	850.749 s	177	6	5	**88.5%**	94%
10	HS + DBOW	150	5	1500	1011.517 s	174	10	4	87%	94%
11	HS + DBOW	150	10	1500	1048.288 s	172	9	6	86%	93.5%

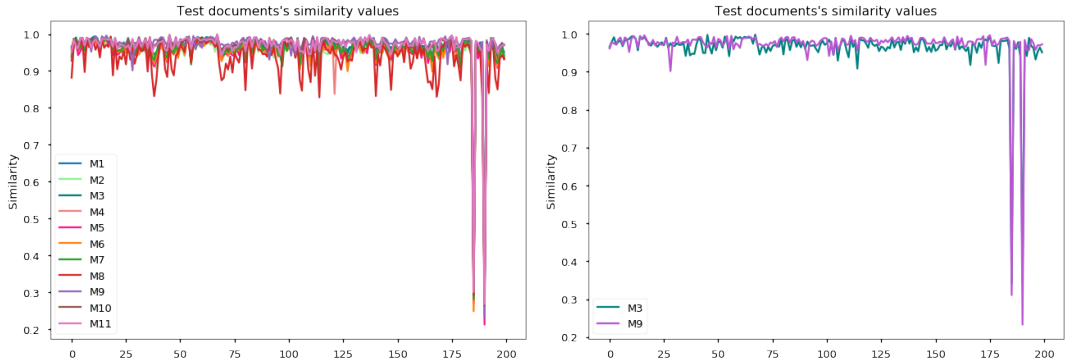

Figure 9. Test Results of Doc2Vec Model.

After all steps of the test, we find that Vector-size 150, Window value 5, and Epochs 1500, is most suitable for the training dataset. It indicates that the Negative Sampling training method is generally superior to the Hierarchical Softmax training method. In the same Negative Sampling training mode, M3 with Distributed Memory (DM) architecture has the highest top-3 recall rate, and its top-1 recall rate performance is resonable, while M9 with Distributed Bag of Words (DBOW) architecture has the highest accuracy (Recall rate@1). To further confirm our parameter selection, we compare the similarity of the 200 documents tested by the two groups (Figure 9), and find that M9 shows a more stable level than M3. Meanwhile, the training time of M9 exhibits strong competitiveness.

4.3. Setup for Human Evaluation

Among the various methods for evaluating the effects of Natural Language Generation (NLG), there are some methods for automatic evaluation, such as NIST, BLEU, and ROUGE. Belz and Reiter [28] believed that automated evaluation methods had great potential after comparing various assessment methods, but the best way to evaluate NLG Models is through human assessment. Meanwhile, our training dataset contained non tagged information, and we could not find a uniform standard or a unified model to evaluate these three models simultaneously.

Since there is not much similarity between E-mails in the dataset (Enron E-mail Dataset), it is difficult to evaluate the functional performance and learning capabilities of the three models. In order to compare the effects of these models, we design the test data according to the training data, following the following rules: change the entity noun (such as time, place and name); change the sentence order of the paragraph; add some information to the E-mail; delete some information from the E-mail; change the expression of the sentence.

We use two criteria to evaluate the performance of three models. In the first case, we compare the final best responses given by the three models, respectively. In other words, we only select the responses of top1 related E-mails extracted by Doc2Vec based and TF-IDF based models, as well as the predicted response generated by the GRU-Sent2Vec hybrid model, to compare the final implementation results of our experiment. In the second case, we only compare the two information retrieval models. After the first four rounds of training on new test E-mails, the fifth designed E-mail will enter into the two models as a query. The experimental results of the two models may extract four similar E-mails with different performance. We will list the top five similar E-mails extracted by the two models respectively with their corresponding similarity scores, and then five participants from different academic fields will choose the model which gave the suggestion that matches the similarity most closely based on subjective judgement.

4.4. SEMS Client Implementation

To display the results and analyse the experimental effect intuitively, we designed a simple client for visualisation. Users can select either to reply directly or to use or modify intelligent suggestions. If the latter is chosen, by clicking the button on the robot in Figure 10 three suggestion responses from each of the three models are presented.

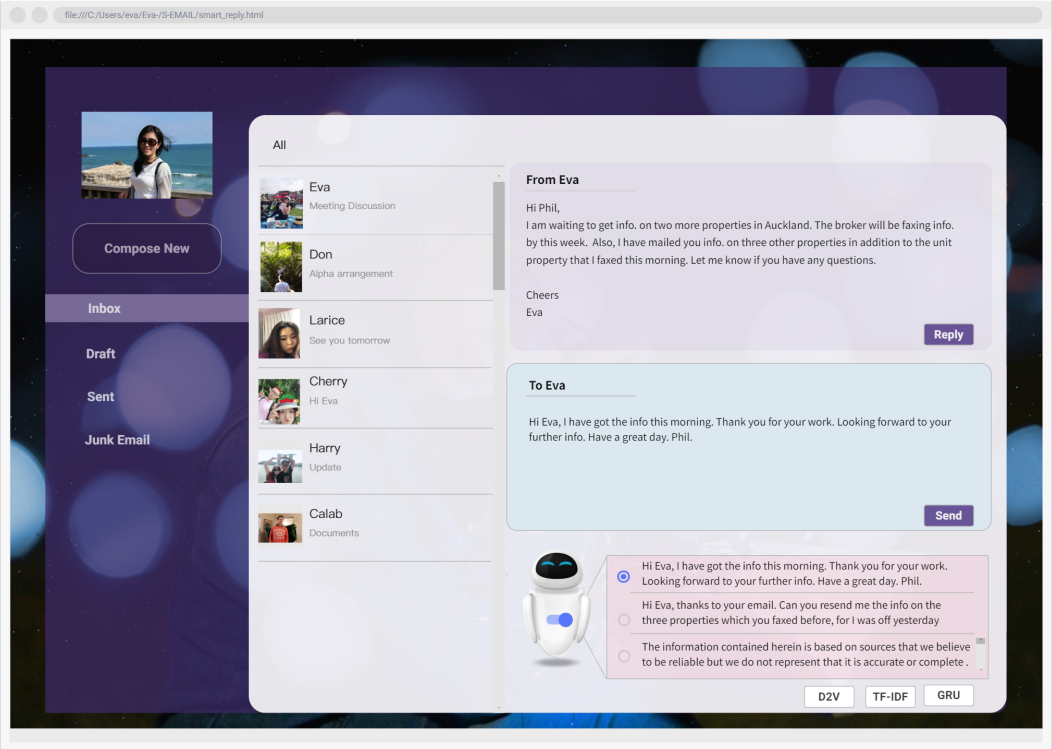

Figure 10. Intelligent E-mail Client—Default Mode. The image of the robot is adopted from the movie WALL-E by Pixar.

4.5. Results and Discussion

4.5.1. Comparison of the Three Models

Figure 11 shows an example that is taken from topic 1 of the experimental results from the three models. As we mentioned earlier, these three models use two different methods. Therefore, in the first stage of human evaluation, we make subjective selections on the final response suggestions, which are also the final results of our experiment. We select the top 1 response from two information

retrieval models as their best response suggestions. Also, the one predictive response generated by the GRU-Sent2Vec hybrid model is treated as the object of evaluation for this model.

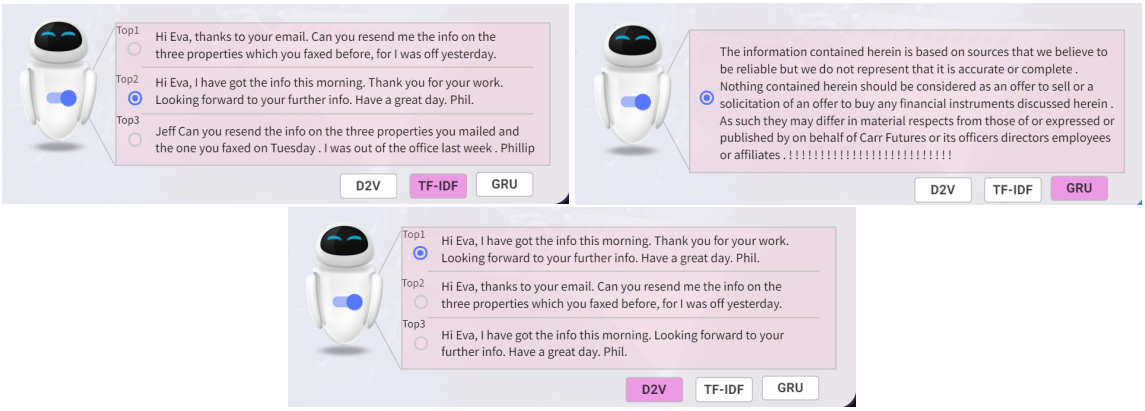

Figure 11. Three Models Results (The new received Email is: "Hi Phil, I am willing to get info. of two more properties in Auckland. The broker will be faxing info. by this week. Also, I have mailed you the info. on three other properties in addtion to the unit property that I faxed this morning. Let me know if you have any questions. cheers Eva").

After several rounds of training, the three models each suggested responses to five new E-mails. The first round of the evaluation of the overall result is shown in Figure 12. According to the subjective evaluation, five participants chose relevant answers for each new E-mail, and this was a multiple-choice process. The results show that TF-IDF and Doc2Vec, the two information retrieval models, had better performance than the information generation model.

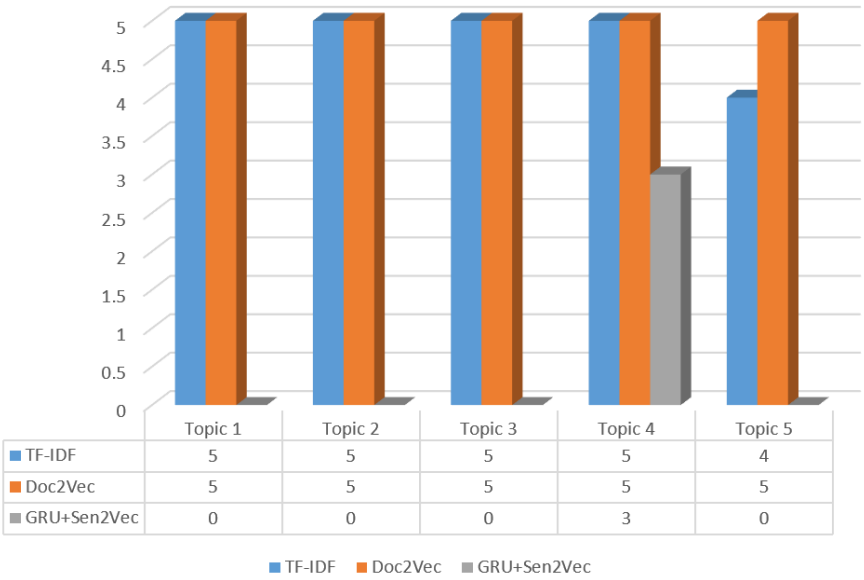

Figure 12. Human Evaluation Results.

The GRU-Sent2Vec hybrid model is not ideal for several reasons. Although we considered this result at the beginning, the main reason is the quality of our training corpus. For deep neural networks, learning relatively accurate feature rules first requires vast datasets. The total number of sentences in our training dataset is 99,431, which is not sufficient. Second, the average number of repeated sentences in the training dataset is only 2.3 (Figure 13), while the majority of sentences only appear once. Such extremely low probability distribution of repeated sentences can hardly provide adequate learning information for deep neural networks.

```
count    99431.000000
mean         2.344822
std         59.809265
min          1.000000
25%          1.000000
50%          1.000000
75%          2.000000
max      16935.000000
dtype: float64
```

Figure 13. Sentence Repeat Statistics.

We expect that given the right environment, the GRU-Sent2Vec hybrid model can predict and generate a series of sentences as the output according to input sentences. Figure 14 presents its workflow design.

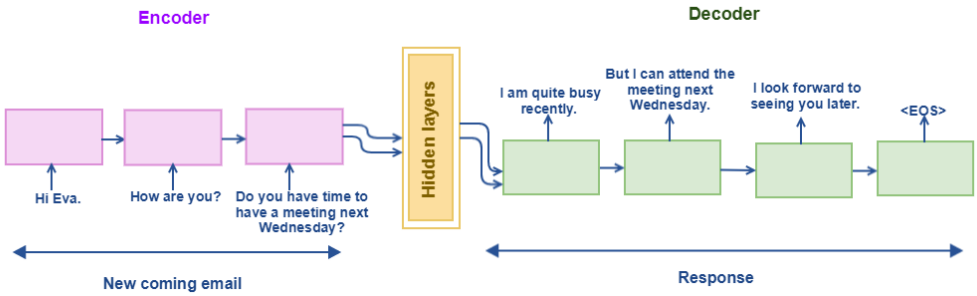

Figure 14. Ideal GRU-Sent2Vec Hybrid Model.

4.5.2. Comparison of the Two Information Retrieval Models

Comparison of Learning Ability: First, we compared the learning ability of the Doc2Vec and TF-IDF Models. Since our training dataset initially did not contain many similar E-mails, we conducted five rounds of training on the models using five similar test E-mails designed and generated for each topic in the previous section. If at the end of each round of training, the model could always find the similar E-mails learned in previous rounds from the database for new E-mails, it meant that the model had the ability to learn new information. The specific process is as follows:

1. We tested it by simply replacing the names of five randomly selected E-mails from the original training dataset. Both models found the five related original E-mails in the first most similar ranking.
2. We put the first round of E-mails together with the responses of corresponding designs into the training set, and then we modified the five selected topic E-mails by changing the order of the sentences. In the second round of testing, the two models found 10 related E-mails in the first two most similar rankings including the original 5 E-mails and the 5 E-mails put into the training set after the first round.
3. We put the E-mails from the second round of E-mail modification into the training set together with the replies from the corresponding designs, and then we modified the 5 selected topic E-mails

by adding some information. The results of the third round of testing showed that in the first 3 most similar rankings, TF-IDF found 14 related E-mails while Doc2Vec found 15 related E-mails with the original 5 E-mails and 10 E-mails put into the training set after the first two rounds.

4. We put the E-mails from the third round of E-mail modification into the training set together with the replies from the corresponding designs, and then we modified the 5 selected topic E-mails by deleting some information. In the fourth round of tests, TF-IDF found 18 related E-mails in the first 5 most similar rankings, while Doc2Vec found 19 related E-mails involving in the original 5 E-mails and 15 E-mails put into the training set after the first three rounds.
5. We put the E-mails from the fourth round of design into the training set together with the replies from the corresponding designs, and then we modified the 5 selected topic E-mails by changing the expression of some information. The results of the fifth round of testing showed that in the first 4 most similar rankings, TF-IDF found 19 related E-mails and Doc2Vec found 23 related E-mails with the original 5 E-mails and the 20 E-mails put into the training set after the first four rounds.

After five rounds of training, the results of each round could reflect the models' learning abilities. The Doc2Vec model presented a more stable learning ability, especially in the fifth round. When the semantic expression was changed, it presented a better information resolution ability than the TF-IDF model. It was proven that Doc2Vec is capable of extracting semantic correlations between words in an article. The results are shown in Figure 15.

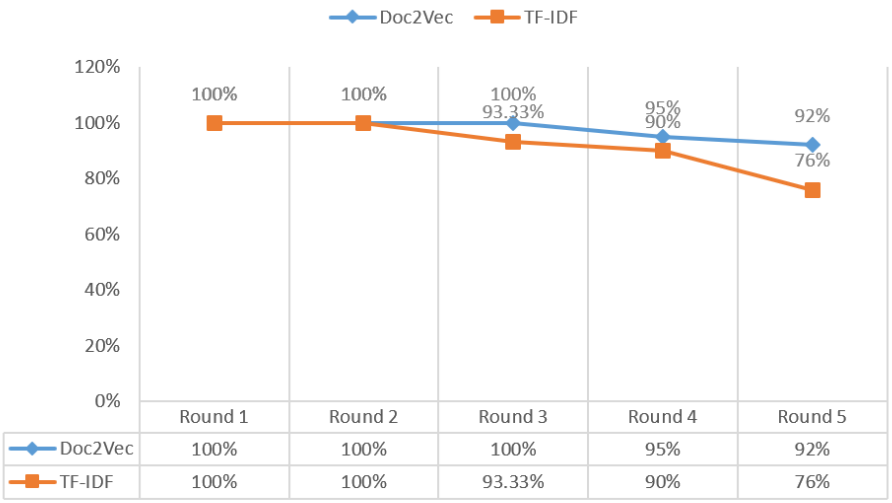

Figure 15. Learning Ability of TF-IDF and Doc2Vec.

Comparison of Effect: After five rounds of training, five participants subjectively evaluated the accuracy of the two models based on the information retrieval mechanism and compared the similarity score. Figure 16 presents five new test E-mails on behalf of the five topics. The top five E-mails with the highest similarity in each topic were extracted from the training dataset by two models. Meanwhile, five participants compared and selected the results extracted from the two models under five topics, which means there are twenty-five results that should be selected for each topic, which was a single selection process.

From the results of the selection data given by the five participants, we concluded that the effect of the Doc2Vec model was significantly better than that of the TF-IDF model from the perspective of subjective evaluation.

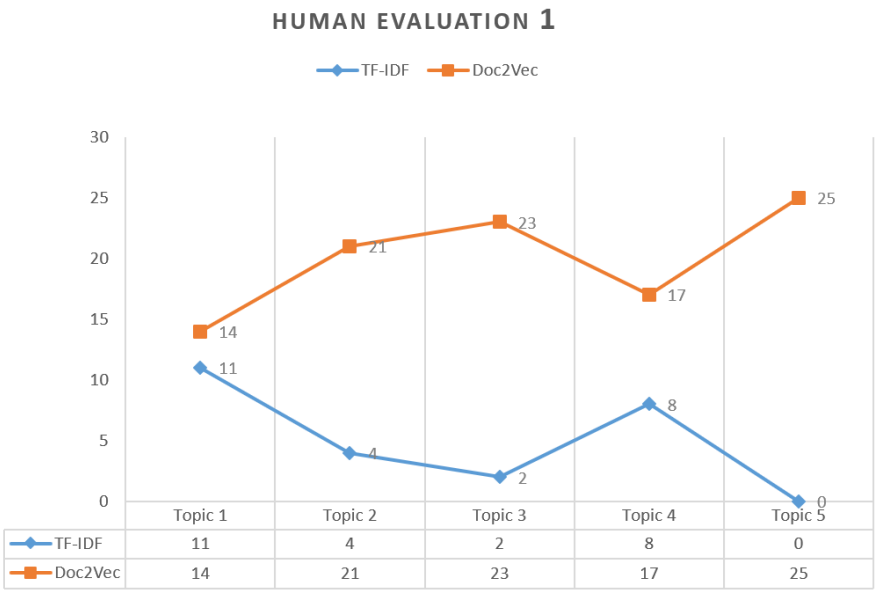

Figure 16. Effect Comparison Between TF-IDF and Doc2Vec.

5. Conclusions

This paper presents a Smart Email Management System (SEMS), a software application and design solution. SEMS is based on machine learning and deep learning technologies, which improves the effectiveness of people who are required to handle the practical problems of using E-mail in daily life and corporate business. The first thing to be emphasised is that we designed a novel GRU-Sent2Vec hybrid model that can be used to predict responses based on sentence-level, which goes beyond the limits of word-level prediction. Although the quality of the training corpus limits the effect, the results have been informative and have the potential to guide further development for industrial applications in the future. As for the evaluation of the effects, it has a subjective component. We designed a set of human evaluation questionnaires. The results of the questionnaire survey showed that this research project has significant application value.

Our experimental models gave coherent and reasonable responses after analysing newly received E-mails. However, there are still many limitations and challenges that we have plan to address in future. The first limitation is the training dataset. Unlike E-mail datasets from help desk or customer service centres, which contain many similar inquiries from customers, our experimental dataset has very little similarity between E-mails because the E-mails were collected from Enron employees. In future we plan to approach some customer service centres to enrich our dataset. The second limitation is the computing power for achieving machine learning or deep learning algorithms. The cost for training our models especially GRU was very high. In future we have a plane to minimise our training cost.

Author Contributions: Y.F. developed the methodology and implemented the design of the research. M.A.N. conceptualised the idea and formulated the research goals and aims. F.M. administrated the project and coordinated responsibility for the research activities planning and execution. A.T. reviewed and polished the draft of the paper. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Acknowledgments: The research is technically supported by National University of Computer & Emerging Sciences, Pakistan and Auckland University of Technology, Auckland, New Zealand.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. The Radicati Group. *Email Statistics Report, 2018–2022-Executive Summary*; Radicati Group, Inc.: Palo Alto, CA, USA, 2018; p. 3.
2. Coussement, K.; Van den Poel, D. Improving customer complaint management by automatic email classification using linguistic style features as predictors. *Decis. Support Syst.* **2008**, *44*, 870–882. [CrossRef]
3. Whittaker, S.; Sidner, C. Email overload: Exploring personal information management of email. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, Vancouver, BC, Canada, 13–18 April 1996; pp. 276–283.
4. Dabbish, L.A.; Kraut, R.E. Email overload at work: An analysis of factors associated with email strain. In Proceedings of the 2006 20th Anniversary Conference on Computer Supported Cooperative Work, Banff, AB, Canada, 4–8 November 2006; pp. 431–440.
5. Pazos, P.; Chung, J.M.; Micari, M. Instant messaging as a task-support tool in information technology organizations. *J. Bus. Commun.* **2013**, *50*, 68–86. [CrossRef]
6. Hair, M.; Renaud, K.V.; Ramsay, J. The influence of self-esteem and locus of control on perceived email-related stress. *Comput. Hum. Behav.* **2007**, *23*, 2791–2803. [CrossRef]
7. Linggawa, I. Reusing Past Replies to Respond to New Email: A Case-Based Reasoning Approach. Ph.D. Thesis, Auckland University of Technology, Auckland, New Zealand, 2017.
8. Dredze, M.; Brooks, T.; Carroll, J.; Magarick, J.; Blitzer, J.; Pereira, F. Intelligent email: Reply and attachment prediction. In Proceedings of the 13th International Conference on Intelligent User Interfaces, Gran Canaria, Spain, 13–16 January 2008; pp. 321–324.
9. Yang, L.; Dumais, S.T.; Bennett, P.N.; Awadallah, A.H. Characterizing and predicting enterprise email reply behavior. In Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, Tokyo, Japan, 7–11 August 2017; pp. 235–244.
10. Tsay-Vogel, M.; Shanahan, J.; Signorielli, N. Social media cultivating perceptions of privacy: A 5-year analysis of privacy attitudes and self-disclosure behaviors among Facebook users. *New Med. Soc.* **2018**, *20*, 141–161. [CrossRef]
11. Lapalme, G.; Kosseim, L. Mercure: Towards an automatic e-mail follow-up system. *IEEE Comput. Intell. Bull.* **2003**, *2*, 14–18.
12. Zhu, X.; Li, T.; De Melo, G. Exploring semantic properties of sentence embeddings. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), Melbourne, Australia, 15–20 July 2018; pp. 632–637.
13. Schank, R.C. *Dynamic Memory: A Theory of Reminding and Learning in Computers and People*; Cambridge University Press: Cambridge, UK, 1982; Volume 240.
14. Hewlett, W.R.; Freed, M. An email assistant that learns to suggest reusable replies. In *AAAI Workshop, Technical Report WS-08-04*; AAAI: Palo Alto, CA, USA, 2008; pp. 28–35.
15. Kannan, A.; Kurach, K.; Ravi, S.; Kaufmann, T.; Tomkins, A.; Miklos, B.; Corrado, G.; Lukacs, L.; Ganea, M.; Young, P.; et al. Smart reply: Automated response suggestion for email. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016; pp. 955–964.
16. Giles, C.L.; Kuhn, G.M.; Williams, R.J. Dynamic recurrent neural networks: Theory and applications. *IEEE Trans. Neural Netw.* **1994**, *5*, 153–156. [CrossRef]
17. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef] [PubMed]
18. Parameswaran, A.; Mishra, D.; Bansal, S.; Agarwal, V.; Goyal, A.; Sureka, A. Automatic Email Response Suggestion for Support Departments within a University. *Peerj* **2018**, *6*, e26531v1. [CrossRef]
19. Bengio, Y.; Ducharme, R.; Vincent, P.; Jauvin, C. A neural probabilistic language model. *J. Mach. Learn. Res.* **2003**, *3*, 1137–1155.
20. Trstenjak, B.; Mikac, S.; Donko, D. KNN with TF-IDF based Framework for Text Categorization. *Procedia Eng.* **2014**, *69*, 1356–1364. [CrossRef]
21. Yang, J.; Jiang, Y.G.; Hauptmann, A.G.; Ngo, C.W. Evaluating bag-of-visual-words representations in scene classification. In Proceedings of the International Workshop on Workshop on Multimedia Information Retrieval, Augsburg, Germany, 24–29 September 2007; pp. 197–206.

22. Kim, D.; Seo, D.; Cho, S.; Kang, P. Multi-co-training for document classification using various document representations: TF–IDF, LDA, and Doc2Vec. *Inf. Sci.* **2019**, *477*, 15–29. [CrossRef]
23. Le, Q.; Mikolov, T. Distributed representations of sentences and documents. In Proceedings of the International Conference on Machine Learning, Bejing, China, 22–24 June 2014; pp. 1188–1196.
24. Mikolov, T.; Chen, K.; Corrado, G.; Dean, J. Efficient estimation of word representations in vector space. *arXiv* **2013**, arXiv:1301.3781.
25. Mikolov, T.; Deoras, A.; Kombrink, S.; Burget, L.; Černockỳ, J. Empirical evaluation and combination of advanced language modeling techniques. In Proceedings of the Twelfth Annual Conference of the International Speech Communication Association, Florence, Italy, 27–31 August 2011.
26. Vinyals, O.; Le, Q. A neural conversational model. *arXiv* **2015**, arXiv:1506.05869.
27. Chung, J.; Gulcehre, C.; Cho, K.; Bengio, Y. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv* **2014**, arXiv:1412.3555.
28. Belz, A.; Reiter, E. Comparing automatic and human evaluation of NLG systems. In Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics, Trento, Italy, 3–7 April 2006.