Volume 12 ▪ Issue 2 ▪ June 2021

INTERNATIONAL JOURNAL OF COMPUTATIONAL LINGUISTICS (IJCL)

Publication Frequency: 6 Issues / Year

ISSN : 2180-1266

CSC PUBLISHERS
http://www.cscjournals.org

INTERNATIONAL JOURNAL OF COMPUTATIONAL LINGUISTICS (IJCL)

VOLUME 12, ISSUE 2, 2021

EDITED BY
DR. NABEEL TAHIR

ISSN (Online): 2180 - 1266

International Journal of Computational Linguistics (IJCL) is published both in traditional paper form and in Internet. This journal is published at the website https://www.cscjournals.org, maintained by Computer Science Journals (CSC Journals), Malaysia.

IJCL Journal is a part of CSC Publishers
Computer Science Journals
https://www.cscjournals.org

INTERNATIONAL JOURNAL OF COMPUTATIONAL LINGUISTICS (IJCL)

Book: Volume 12, Issue 2, June 2021

Publishing Date: 30-06-2021

ISSN (Online): 2180-1266

IJCL Journal is a part of CSC Publishers
https://www.cscjournals.org

Published in Malaysia

Typesetting: Camera-ready by author, data conversation by CSC Publishing Services – CSC Journals, Malaysia

CSC Publishers, 2021

TABLE OF CONTENTS

Volume 12, Issue 2, June 2021

Pages

EDITORIAL PREFACE

The International Journal of Computational Linguistics (IJCL) is an effective medium for interchange of high quality theoretical and applied research in Computational Linguistics from theoretical research to application development. This is the *Second* Issue of Volume *Twelve* of IJCL. The Journal is published bi-monthly, with papers being peer reviewed to high international standards. International Journal of Computational Linguistics (IJCL) publish papers that describe state of the art techniques, scientific research studies and results in computational linguistics in general but on theoretical linguistics, psycholinguistics, natural language processing, grammatical inference, machine learning and cognitive science computational models of linguistic theorizing: standard and enriched context free models, principles and parameters models, optimality theory and researchers working within the minimalist program, and other approaches.

IJCL give an opportunity to scientists, researchers, and vendors from different disciplines of Artificial Intelligence to share the ideas, identify problems, investigate relevant issues, share common interests, explore new approaches, and initiate possible collaborative research and system development. This journal is helpful for the researchers and R&D engineers, scientists all those persons who are involve in Computational Linguistics.

Highly professional scholars give their efforts, valuable time, expertise and motivation to IJCL as Editorial board members. All submissions are evaluated by the International Editorial Board. The International Editorial Board ensures that significant developments in image processing from around the world are reflected in the IJCL publications.

IJCL editors understand that how much it is important for authors and researchers to have their work published with a minimum delay after submission of their papers. They also strongly believe that the direct communication between the editors and authors are important for the welfare, quality and wellbeing of the Journal and its readers. Therefore, all activities from paper submission to paper publication are controlled through electronic systems that include electronic submission, editorial panel and review system that ensures rapid decision with least delays in the publication processes.

To build its international reputation, we are disseminating the publication information through Google Books, Google Scholar, J Gate, ScientificCommons, Scribd, CiteSeerX Docstoc and many more. Our International Editors are working on establishing ISI listing and a good impact factor for IJCL. We would like to remind you that the success of our journal depends directly on the number of quality articles submitted for review. Accordingly, we would like to request your participation by submitting quality manuscripts for review and encouraging your colleagues to submit quality manuscripts for review. One of the great benefits we can provide to our prospective authors is the mentoring nature of our review process. IJCL provides authors with high quality, helpful reviews that are shaped to assist authors in improving their manuscripts.

Editorial Board Members
International Journal of Computational Linguistics (IJCL)

Developing AI Tools For A Writing Assistant: Automatic Detection of dt-mistakes In Dutch

Wouter Mercelis *wouter.mercelis@kuleuven.be*
Faculteit Letteren/Onderzoekseenheid Taalkunde/Onderzoeksgroep
Kwantitatieve Lexicologie en Variatielinguïstiek (QLVL)
KU Leuven, Leuven, 3000, Belgium

Abstract

This paper describes a lightweight, scalable model that predicts whether a Dutch verb ends in -d, -t or -dt. The confusion of these three endings is a common Dutch spelling mistake. If the predicted ending is different from the ending as written by the author, the system will signal the dt-mistake. This paper explores various data sources to use in this classification task, such as the Europarl Corpus, the Dutch Parallel Corpus and a Dutch Wikipedia corpus. Different architectures are tested for the model training, focused on a transfer learning approach with ULMFiT. The trained model can predict the right ending with 99.4% accuracy, and this result is comparable to the current state-of-the-art performance. Adjustments to the training data and the use of other part-of-speech taggers may further improve this performance. As discussed in this paper, the main advantages of the approach are the short training time and the potential to use the same technique with other disambiguation tasks in Dutch or in other languages.

Keywords: NLP, Dutch, AI, Spelling Correction, Transfer Learning.

1 INTRODUCTION

1.1 Goal

This paper describes a machine learning approach to predicting whether a Dutch verb ends in *-d*, *-t* or *-dt*. The first section provides background information about this topic. The following sections give an overview of the used data sets and the different models that were used during training. The last section discusses strong and weak points of this paper's approach.[1] The methodology is deductive, as I started from a theoretic point of view (fast and lightweight neural networks) and brought this into practice. Real mistakes from students were used for analysis purposes.

The paper can be linked to other entries in the International Journal of Computational Linguistics, regarding spell checkers [1], verb analysis [2] and stemming algorithms [3].

1.2 Grammatical Error Detection

Most of the work in the NLP field regarding grammatical error detection focuses on general systems and on determining whether or not a sentence is grammatical. Examples include Liu & Liu [4] and Li et al. [5]. Rather than designing an all-purpose grammatical error detection system, this project aims to address one specific problem. An additional complicating factor is that most of the work undertaken is tailored to the specifics of English. The few studies on Dutch do not adopt the general approach; instead, they focus on one specific problem, similar to this article. Heyman

[1] This project was undertaken against the backdrop of an internship at Reimagine, a Brussels-based AI start-up. Because of this, the code used during the project cannot be made available, being property of the company. The test data was provided by ILT (Instituut voor Levende Talen), which commissioned the project as future part of a writing assistant for secondary school students. . I would like to thank my colleagues at Reimagine, especially Ferre Jacobs and Toon Lybaert, who provided useful insights for this article. I would also like to thank Alek Keersmaekers, Toon Van Hal and Liesbeth Augustinus for their proofreading and valuable guidance.

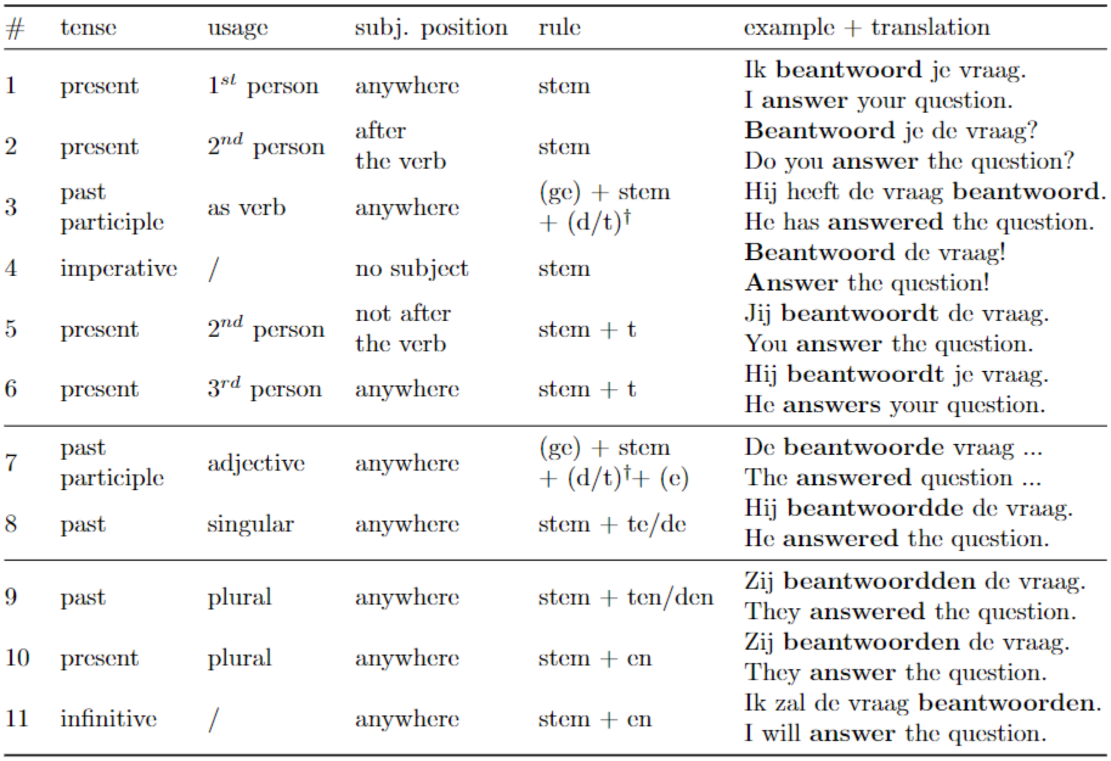
et al. [6] also worked on dt-mistakes, while Allein et al. [7] provided a model that disambiguates *die* and *dat*. Dt-mistakes will be discussed in more detail in the next section.

The first grammatical error detection systems built were rule-based. However, writing such manual rules is very time-consuming, does not generalise well and does not capture more complex phenomena like long-distance dependencies [8]. Up until recently, a statistical approach with n-grams in large corpora was used to solve this issue. Data sparsity forces the n-gram approach to work with large data sets such as the Google n-gram corpus [9]. These models need to run on expensive high processing computing systems due to their large size. Additionally, the n-gram approach still has problems capturing more complex phenomena like long-distance dependencies [8]. Heyman et al. [6] referred to an approach in which a word-based classifier is trained per word pair that can give way to a dt-mistake (e.g. *gebeurd* and *gebeurt*). However, this approach does not generalise well, as the system does not learn the actual dt-rule.

Nowadays, neural network models are considered to be better equipped to perform such error detection tasks. These models have the capability to model complex sentences containing, for example, many long-term dependencies [10].

1.3 The Dutch dt-rule

Dutch regular verbs adhere to the following conjugation rules, as illustrated in Table 1. In this table, the first six rules are considered dt-rules in a narrow sense. Rules 1-11 can be seen as the dt-rules in a broader sense, as they involve endings other than just *-d*, *-t* and *-dt*.



#	tense	usage	subj. position	rule	example + translation
1	present	1^{st} person	anywhere	stem	Ik **beantwoord** je vraag. I **answer** your question.
2	present	2^{nd} person	after the verb	stem	**Beantwoord** je de vraag? Do you **answer** the question?
3	past participle	as verb	anywhere	(ge) + stem + (d/t)†	Hij heeft de vraag **beantwoord**. He has **answered** the question.
4	imperative	/	no subject	stem	**Beantwoord** de vraag! **Answer** the question!
5	present	2^{nd} person	not after the verb	stem + t	Jij **beantwoordt** de vraag. You **answer** the question.
6	present	3^{rd} person	anywhere	stem + t	Hij **beantwoordt** je vraag. He **answers** your question.
7	past participle	adjective	anywhere	(ge) + stem + (d/t)†+ (e)	De **beantwoorde** vraag ... The **answered** question ...
8	past	singular	anywhere	stem + te/de	Hij **beantwoordde** de vraag. He **answered** the question.
9	past	plural	anywhere	stem + ten/den	Zij **beantwoordden** de vraag. They **answered** the question.
10	present	plural	anywhere	stem + en	Zij **beantwoorden** de vraag. They **answer** the question.
11	infinitive	/	anywhere	stem + en	Ik zal de vraag **beantwoorden**. I will **answer** the question.

TABLE 1: An overview of the Dutch dt-rules, cited from Heyman et al. [6].

In Dutch, *-d* at the end of a word is pronounced as unvoiced *-t*, which explains why there is no audible distinction between words such as *rat* ('rat') and *rad* ('wheel'). When the stem of the verb ends in *-d*, confusion arises. The first person singular ends in *-d*, e.g. *ik word* ('I become'), but in the second and third person singular present, the ending *-t* is added to the stem, e.g. *hij wordt* ('he becomes')[2]. The *-dt* at the end of a word, which occurs when a stem ending in *-d* receives the verb ending *-t*, also has an unvoiced pronunciation. Verhaert & Sandra [11] argued that the root cause of dt-mistakes is so-called homophone dominance. This means that the writer will choose the most frequent form of a verb, when put under stress or when distracted.

Another cause underlying many dt-mistakes is the inversion rule for second person singular. Normally, this point of view has an ending in *-t*, but when the sentence is inversed (e.g. in a question), there is no such ending. This can be illustrated with examples 1 and 2 [12]:

1. *Jij beantwoordt haar vraag.*
   You answer her question.'
2. *Beantwoord je haar vraag?*
   'Do you answer her question?'
3. *Beantwoordt je moeder haar vraag?*
   'Does your mother answer her question?'

Furthermore, the second person singular personal pronoun *jij* also has the phonetically weakened form *je*. However, the second person singular possessive pronoun *jouw* has the same phonetically weakened form *je*, e.g. *jouw/je moeder* ('your mother'). In non-inverted sentences, this does not cause any problems, as the endings for the second and third person singular are the same. However, in inverted sentences such as questions, the ending differs. This is illustrated in examples 2 and 3 [12].

A third cause of dt-mistakes lies in the past participle, which is regularly formed by adding a prefix *'ge-'* to the verb stem and a suffix. The suffix is *-t* when the verb stem ends in an unvoiced consonant, but it is *-d* when the verb stem ends in a voiced consonant. For example, the verb *suizen*, meaning 'to whiz', does not have the predicted stem *'suiz'*; rather, it has the stem *'suis'*. The underlying stem *'suiz'* is used to determine the ending, resulting in the participle *gesuisd* and not *'gesuist'* [12].

Another category of participle-based dt-mistakes occurs when the past participle is irregularly formed. For example, when a verb stem already starts with *'ge-'*, the past participle prefix *'ge-'* is dropped. When the verb stem ends in a voiced consonant, the participle ends in *-d*, while the second and third person singular present forms end in *–t* [12].

4. *Hij getuigt tegen mij.*
   'He testifies against me.'
5. *Hij heeft tegen mij getuigd.*
   'He has testified against me.'

Finally, the difference between context-dependent and context-independent mistakes needs to be stressed. All previous examples are prone to context-dependent mistakes, as the different forms of the verb all exist in Dutch but in different contexts. However, it is also possible to make a context-independent mistake and create a non-existent form. An example from the test data is *hij duwd*, a non-existent form, instead of *hij duwt* ('he pushes').

1.4 Previous Work by Heyman et al. [6]

Heyman et al. [6] focused on context dependent dt-mistakes, while this paper aims to cover both context dependent and context independent errors. While it is theoretically possible to use a dictionary lookup for these context independent mistakes, it is convenient that the model is able to predict these as well, as this reduces the amount of memory needed.

Another difference is that Heyman et al. created a system to introduce mistakes in a text to later correct them. In this paper, for the training data, we have used data without mistakes, as the

[2] Phonetically, both forms can be transcribed as follows: /ʋɔrt/

model only trains on predicting the right ending. For the test data, we did not insert mistakes in a text with an algorithm. Instead, I searched for texts containing actual mistakes and focused on sentences written by students. If the model's prediction was different from a student’s written verb ending, a signal was given that the form may be incorrect.

The architecture of the models also differs. This paper makes use of the ULMFiT approach with an AWD-LSTM network, which is a regular LSTM network with tweaked hyperparameters. Heyman et al. used a custom engineered setup, consisting of a context encoder and a verb encoder, concatenated in a feed-forward neural network (FFNN). The final probability is computed using a softmax function over the FFNN-layer. A visualisation of Heyman et al.'s model is shown in Figure 1. Another difference between the two approaches lies in the use of part-of-speech taggers. Heyman et al. used TreeTagger [13], while I used spaCy's PoS tagger [14]. This decision was primarily made because the commissioning institute (ILT) already works with spaCy, thus making the final implementation easier. It is possible that spaCy's PoS tagger missed some verbs, which are thus not present in the data. In the meantime, the model aimed to accurately predict the ending of the detected verbs. In this implementation, a verb was considered correct if the written ending was the same as the model’s predicted ending. Otherwise, the verb was considered to be incorrect.

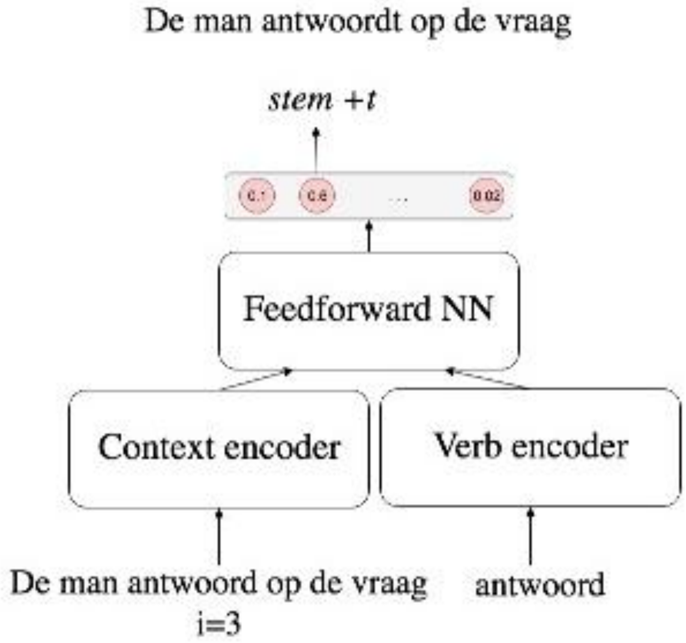


FIGURE 1: A visualisation of the model used by Heyman et al. [6].

As mentioned above, there is a difference between dt-mistakes in the narrow sense (confusing -*d*, -*t* and -*dt*), and dt-mistakes in the broad sense (adding -*de*, -*dde(n)*, -*te*, -*tte(n)* etc.). Heyman et al. [6], working on the broad sense, experienced some trouble with the multitude of labels, causing them to drop infrequent labels such as the past plural forms (number 9 in Table 1). As mistakes in the narrow sense are in absolute numbers much more frequent than mistakes in the broad sense, we decided to focus on the core task of discerning -*d*, -*t* and -*dt*. Furthermore, the broad sense is harder to grasp completely, as some minimal pairs show the doubling of vowels as well (e.g. *vergrote*, *vergrootte* ('enlarged')). However, it should be possible to train another, separate model that corrects all possible mistakes in the broad sense. We leave that problem for further research.

2 DATA

2.1 Training Data

Three main data sets were used for training: the Europarl corpus, the Dutch Parallel Corpus (DPC) and a corpus made from Dutch Wikipedia articles. The development data used during the training of the models stemmed from the random selection of 20% of the sentences in the training data. A detailed description of the data is included in the appendix in Table 5.

2.1.1 Europarl Corpus

The Europarl corpus [15] collects the proceedings of the European Parliament and is available in several languages, including Dutch. Professionals created this corpus, which means that the quality of the text data is assured. However, the domain variation is limited, both in content, as only politics are covered, and on a grammatical level, as there are few verbs found in the second person. To filter out a first wave of sentences with no verbs ending in *-d*, *-t* or *-dt* at all, I used a subset of Heyman et al.'s [6] Europarl corpus in which each sentence contains at least one verb ending subject to the Dutch dt-rule in its broad sense.

I altered the data set by evening out the counts of *-d*, *-t* and *-dt*. As *-dt* occurs less than *-d* and *-t* (between 5% and 10% of the cases, depending on the data set), it was difficult for the model to predict the dt-cases. This data alteration was also performed on the other two corpora, described infra.

2.1.2 Dutch Parallel Corpus

The Dutch Parallel Corpus (DPC) [16] is a multilingual parallel corpus. It consists of parallel texts in Dutch, English and French. The main goal of the corpus is to provide material for multilingual tasks such as machine translation, but it is also possible to use a corpus containing only one language. An advantage of the DPC is that it covers a wide range of domains, although the total size of the corpus is significantly smaller than the Europarl corpus.

2.1.3 Dutch Wikipedia Corpus

As using a complete Dutch Wikipedia dump [17] would result in too much data, I selected articles at random to create a subset of the entire corpus. The advantage of the Wikipedia corpus is that all texts are written in a (pseudo-)scientific style. As the goal of the study was to improve a tool correcting student-written texts in a similar style, this data set closely fit the expectations. A disadvantage of the Wikipedia corpus is that all the sentences are written in third person, while students use the first person when writing about their personal experiences.

2.1.4 Combinations of Data Sets

To circumvent problems such as the absence of first person pronouns in the Wikipedia corpus, I decided to combine data sets. The Wikipedia corpus was used as the base corpus, with the addition of either the DPC or the Europarl corpus. Using the equalised version ensured that the resulting corpus was small enough to be computationally efficient. Moreover, it guaranteed that there were enough dt-cases, while still maintaining the difference in usage between *-d*, *-t* and *-dt*.

In theory, combining all three data sets would also be possible. As the training time would further increase, and the current results turned out to be satisfactory, I did not train a model combining all three data sets.

2.2 Test Data

Over the course of the study, I added several sets of test data, taken from various sources. Some data sets were added to compare results with other research, while others were added to approximate real world data. The data sets are summarised in Table 2.

2.2.1 Test Corpus from Heyman et al. [6]

The test set containing online dt-quizzes was made publicly available [18]. For this paper, sentences not relating to dt-mistakes in the narrow sense were removed. This makes an accurate comparison with Heyman et al. difficult, as I do not have access to their results on this particular subset.

2.2.2 Scholieren.com

Scholieren.com is a website where students (mainly from secondary school) can upload their writings (book reports, essays etc.). Other users can rate the uploaded writings, but the writings are not checked for spelling mistakes or factual errors. The writings uploaded to this website are thus highly approximate to the target group of secondary school students. Four texts were chosen, written by Flemish students of various ages, viz. papers written by a first year secondary school student [19], a second year secondary school student [20], a third year secondary school student [21] and a sixth year secondary school student [22], respectively.

From these texts, 100 cases were randomly selected. This is data set Schol_1 in Table 2. Afterwards, I searched for a text with a high number of mistakes. This was done to check whether spaCy's part-of-speech tagger was still able to identify verbs when they were spelled incorrectly. A book review, written by a sixth year secondary school student, meets this criterion [23]. This is data set Schol_2 in Table 2.

2.2.3 Test Set ILT

ILT provided me with their own test set containing actual errors made by students. From this list of 100 sentences, 69 sentences were suitable for d/t/dt-classification. This data set diverges from the pattern seen in the training data, where the ending -dt occurred between 5% and 10% of the cases. In this data set, it is the most frequent ending. A possible explanation is the focus in this data set on the auxiliary verb *worden*, which occurs 28 times. This verb ends in the first person on -d and in the more frequent third person on -dt.

Name	Verb Endings			Total
	d	t	dt	
Heyman	21	16	9	46
Schol_1	45	49	6	100
Schol_2	8	21	4	33
ILT	17	25	27	69
Total	91	111	46	248

TABLE 2: Overview of the used test data sets.

3 MODELS

This section first discusses the preprocessing steps and the main architecture used for the language model. This is followed by an overview of the trained models, after which attention is given to metrics other than accuracy.

3.1 Preprocessing Steps

After downloading the data, the first step consisted of filtering the actual sentences and removing other unnecessary information. For the Wikipedia corpus, this meant deleting titles and subtitles. The DPC, on the other hand, is XML-based. Therefore, the tags allowed me to extract the text and write it to one large text file. For the Europarl corpus, I used the sentence-by-sentence text file from Heyman et al. [6]. As a second step, spaCy's part-of-speech tagger was used for each set of sentences, enabling me to identify the verbs in each sentence. For each verb, the ending (-*d*, -*t* or -*dt*) was cut off and replaced with the dummy symbol #. The actual verb endings were kept to serve as a ground truth for the classifier and to determine whether a mistake was made in the test data. If the verb did not end in -*d*, -*t* or -*dt*, the dummy symbol was still added, and the ending was filled with a slash sign, such as 'is/'.

As there is often more than one verb in a sentence, I decided to repeat sentences in the data set. For each repetition, only one verb was assigned a dummy symbol. The sentence *Ik word opgehaald door mijn moeder,* meaning 'I get picked up by my mother', appears in the data set as:

- Ik wor# opgehaald door mijn moeder.
- Ik word opgehaal# door mijn moeder.

This way, the classifier does not have to predict multiple verbs at the same time, which would be impossible.

Using these steps, the text file was converted to a comma-separated values file (CSV file), with the following columns: sentence ID number, unaltered sentence, sentence with a dummy symbol, verb without an ending and verb ending. The verb without ending column was added to improve the readability of the CSV file. The preprocessing is visualised in Figure 2.

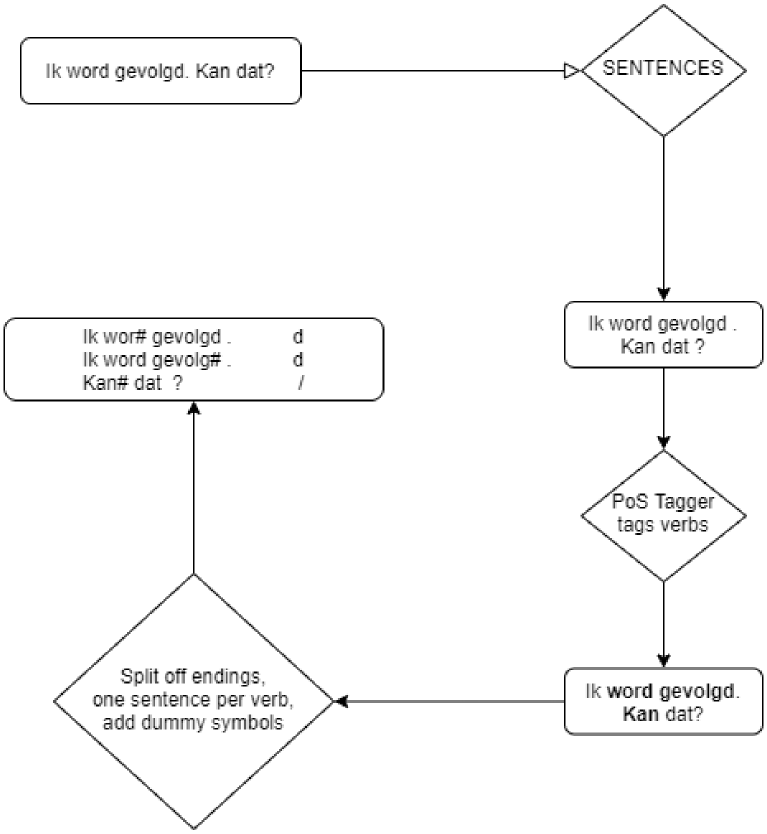


FIGURE 2: A schema highlighting the preprocessing steps.

3.2 Transfer Learning with ULMFiT

3.2.1 ULMFiT

ULMFiT (universal language model fine-tuning for text classification) is a method to introduce inductive transfer learning to various natural language processing tasks. Originally, the technique has found success in Computer Vision, but up until recently this was not the case in NLP due to a lack of knowledge [24]. The model makes use of a three-layer LSTM architecture called AWD-LSTM. This architecture is the same as a regular LSTM (long short-term memory), but uses different dropout hyperparameters. Using an LSTM makes sure that long-term dependencies are covered, as an LSTM is able to filter out important information that spans an entire sentence.

The process behind an ULMFiT model can be divided into three steps. The first step is the acquisition of a general language model. This model can be trained, or an existing model can be used. For this paper, an existing model was used [25], although we briefly considered training our own model. However, the existing model formed a good starting point because it was trained on a large Wikipedia corpus. Nevertheless, fine-tuning these data with our own, smaller Wikipedia corpus increased the performance of the model.

The second step is fine-tuning the general language model on the target task. This takes into account that the training data often have a different distribution than the data used in the general model [24]. It is possible to use discriminative fine-tuning while training the fine-tuned model. This means that different layers of the model are trained with different learning rates, which improves the performance of the model. The vocabulary size of the language model is 60,000 words. The loss function is a standard cross-entropy loss function, which is used for the classifier training as well. This function can be described as follows [26]:

$$loss(x, class) = -\log\left(\frac{\exp(x[class])}{\sum_j \exp(x[j])}\right) = -x[class] + \log\left(\sum_j \exp(x[j])\right)$$

The input of this function is a vector with three elements during the training of the classifier, one for each possible class. The predicted class is positive, while the other two are negative. The displayed result of the loss function is the average loss per epoch. A lower loss means that there is less distance between the predicted classes and the ground truth.

The third and final step involves fine-tuning the classifier. For this task, two linear layers are added to the language model, with a ReLU activation for the first layer and a softmax activation for the second layer. This way, the classifier gives a probability distribution over the classes as output. To fine-tune the classifier, it is possible to use gradual unfreezing. This means that the layers of the model are fine-tuned one by one instead of all layers being fine-tuned at the same time. This process starts at the last layer, as this layer contains the most specific knowledge. By using this technique, the risk of catastrophic forgetting is reduced [24]. A visualisation of the ULMFiT architecture is shown in Figure 3.

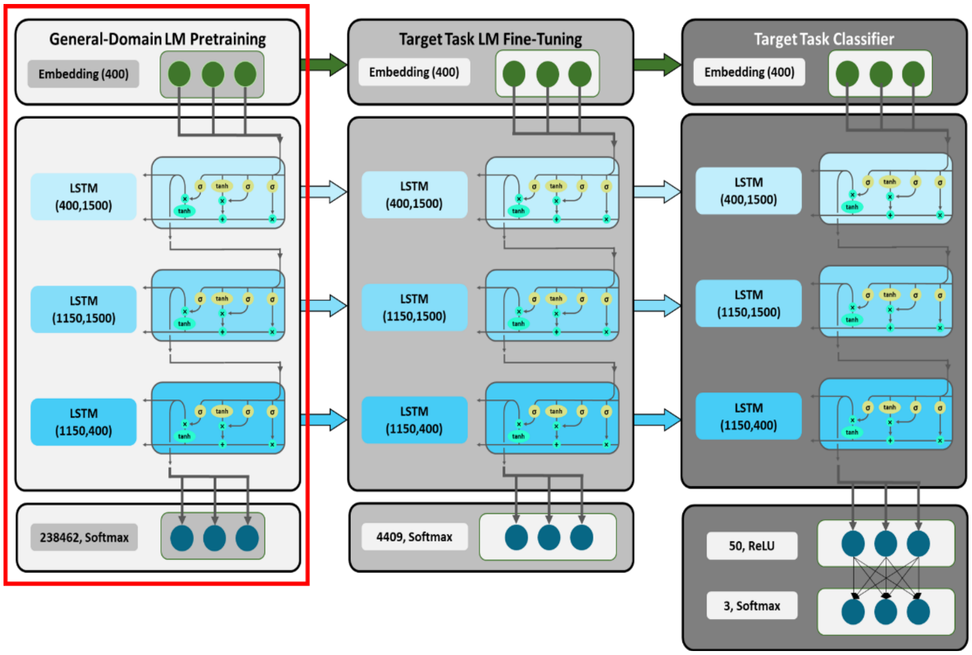


FIGURE 3: A visualisation of the ULMFiT architecture [27].

3.2.2 Fast.ai

Fast.ai is an AI library for Python, built on PyTorch, that aims to make AI more accessible to the general public [28]. As fast.ai v2 [29] was still in development during this study, I used fast.ai v1. The library is built in such a way that both people with limited coding skills and experts are able to work with it. As the developers of fast.ai also developed ULMFiT, the platform is best suited to use this architecture. However, other techniques can be used as well, such as BERT-based models[3].

3.3 Overview of the Trained Models

For this paper, I experimented with various models, whose results will be explained in more detail here, with special attention given to the final model.

The experiments showed that using fast.ai's gradual unfreezing technique provides good results. It predicted the right learning rate per epoch, as shown in the below graph (Figure 4).

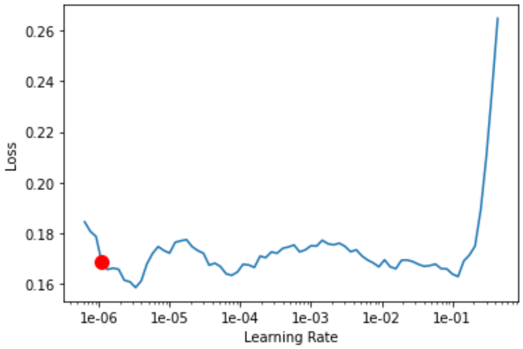


FIGURE 4: The learning rate predictor during model training. The red dot is the suggested learning rate.

Equalising the verbs, as described in section 2.1, resulted in a negative impact. Few training data remained, resulting in a performance hit, especially in the test data.

The final experiments involved the equalised data sets, adding the DPC equalised set to the well-performing Wikipedia corpus in one experiment, and adding the Europarl equalised set to the Wikipedia corpus in another. The combination Wikipedia-Europarl outperformed Wikipedia-DPC by a slight margin, possibly because the Europarl data set was larger. For an unknown reason, the Wikipedia-DPC corpus performs badly on the test data, while the Wikipedia-Europarl combination notes good results. By combining data sets in this way, I found the perfect balance between an unequalised and an equalised data set, combining advantages from both.

The last described model (Wikipedia + equalised Europarl) is my final model, of which the results are highlighted in Table 3. The model was trained in one work day (excluding preprocessing time as this was already done) and is thus easily extendable to other, similar problems. The process of this internship clearly demonstrates that it is important to use data from a domain comparable to the target domain.

The accuracy of the model can still be improved by adding more data containing inverted sentences in the second person singular present. It is not of great importance for the current task, as students are not expected to use this form often in their writings, but it is important if the tool should ever be released for general purposes. Future research can provide a part-of-speech tagger that achieves an even higher accuracy than spaCy's.

3.4 Other Metrics

The main task of this study consisted of training an accurate model. However, there are other metrics that measure whether a model's corrections are right or not, such as precision, recall and

[3] Training BERT-like models did not provide a useful outcome, mainly due to memory issues and the lack of a dummy symbol in the pretrained model.

the F1-score. These metrics are used in the same way as in Heyman et al. [6]. In this section, I will compute these metrics where possible, using the final model on the various test data sets.

As Heyman et al. used different data sets with a classifier that was trained on a slightly different task, it is not possible to make an accurate comparison between their approach and the approach of this paper.

The predicted labels (the verb endings) are divided into four groups:

- True positives (*tp*): the predicted ending corrects the original wrong ending successfully
- False positives (*fp*): the original ending was correct; the predicted ending introduces a mistake
- True negatives (*tn*): both the original and the predicted ending are correct
- False negatives (*fn*): both the original and the predicted ending are incorrect

The precision, recall and F1-score are then defined as follows:

$$prec = \frac{tp}{tp + fp}$$

$$rec = \frac{tp}{tp + fn}$$

$$F_1 = \frac{2prec * rec}{prec + rec}$$

I was not able to measure these metrics for the Heyman data set because we only had the data set with dummy symbols, without actual answers. A different problem was found in the Schol_1 data set, where all the verbs were both correctly spelled and predicted. This way, all the labels belong to the group with true negatives, resulting in a division by zero when calculating precision and recall.

For the Schol_2 data set, calculating these metrics was trivial, as there were mistakes in the original labels and the accuracy of the predictions was 100%. This means that my model achieves a precision, recall and F1-score of 1.

The ILT data set is more interesting for these metrics. The data contains 64 corrected errors, 2 untouched originally correct verbs and 3 mistakes that were not corrected. There were no false positives. This results in a precision of 1, a recall of 0.9552 and an F1-score of 0.9771.

The results using this metrics are listed in Table 4. One should keep in mind that these metrics only consider the verbs in the data set. This means that erroneously written verbs that spaCy's part-of-speech tagger did not consider verbs are not included. However, the tagger still occasionally considered non-existent forms such as *hij duwd* instead of *hij duwt* as verbs. Nevertheless, these wrongly spelled verbs may still form an area of potential improvement.

	Schol_1	**Schol_2**	**ILT**
tp	0	3	64
fp	0	0	0
tn	100	30	2
fn	0	0	3
Precision	/	1	1
Recall	/	1	0.9552
F1	/	1	0.9771

TABLE 4: An overview of the precision, recall and F1-score of the final model, tested on various test data sets. Heyman et al. are not included, as their data consisted of sentences in which the ending should be predicted. Their data did not include writers' filled-in endings with mistakes.

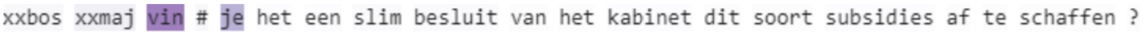
4 DISCUSSION

4.1 General

While the model should be able to generalise well, it has a few disadvantages. For example, the model clearly has trouble predicting the second person singular in inversed sentences. This is due to a lack of training data. As this type of sentence is rather rare in student writings from secondary schools, I chose, due to time constraints, to keep the model as it stands. Needless to say, it is possible to search for more inversed sentences and add them to the training data.

Another possible flaw is the reliance on spaCy's part-of-speech tagger. I observed a case in the test data where a name (*De Gucht*), was incorrectly tagged as a verb, masking the ending *-t*. This can obviously lead to unexpected results. A similar case occurred in the training data, where a sentence contained an enumeration of points, explicitly written as point a until point d. The tagger identified point d as a verb, replacing the ending (the same d) with a dummy symbol. The rest of the verb thus remained empty, causing an unexpected error. Although such mistakes can happen during training, the neural model can handle some noise if the size of the data set is sufficiently large. However, if these mistakes happen when the model is actually in use, the user may not get the expected result. This may occur, for example, if someone uses the name *De Gucht* and the model determines that this is a verb that should end in *–dt*,

It is possible to visualise which words the model deems the most important in order to produce the ending output, thanks to the attention mechanisms. Figure 5 shows the attention the model gives to the sentence *Vind je het een slim besluit van het kabinet dit soort subsidies af te schaffen?* meaning 'Do you think it is a smart decision from the cabinet to abolish subsidies of this kind?' It is clear in the picture that the model gives most of its attention to the verb (without the ending) *'vin'* and to the subject *'je'*. These are indeed the two important factors that determine the verb's ending. In this case, the model correctly predicts the ending *-d*.



```
xxbos xxmaj vin # je het een slim besluit van het kabinet dit soort subsidies af te schaffen ?
```

FIGURE 5: An example of attention: the two important words are highlighted.

The trained model makes use of a dummy symbol to predict verb endings. This means that the model is easy to generalise to other cases and that it can be trained for other classification tasks as well. For example, the diphthong /ɛɪ/ is written in two different ways in Dutch: ei and ij. While a dictionary-based model can solve most of these, as only one of the two possibilities exists, a neural network-based model can be trained to solve the cases where both words exist, based on context. Examples of minimal pairs for this problem are the verbs *leiden* ('to lead') and *lijden* ('to suffer') and the nouns *peil* ('level') and *pijl* ('arrow'). Instead of replacing the verb endings with a dummy symbol, these diphthongs should be replaced in the training data.

Another problem, already studied by Allein et al. [7], is the disambiguation of the Dutch words *die* and *dat*, which was explained in section 1.2. Further research is needed to determine if the model proposed in this report more effectively addresses these specific problems compared to the current state-of-the-art models.

Finally, multi-label classification, such as predicting punctuation, should also be possible. Due to the higher number of labels, this is a harder task, but the setup used during this study can be used here as well.

4.2 Impact and Possible Comparisons with Heyman et al. [6]

The approach in this article differs from the one used by Heyman et al. [6] at several points. Firstly, the structure of the neural network is different, as Heyman et al. [6] used a recurrent neural network, whereas I made use of an LSTM-powered feed forward neural network.

Secondly, this article does not cover the same set of spelling mistakes as Heyman et al. [6], as they tried to detect dt-mistakes in the broad sense, while this article focuses on dt-mistakes in the narrow sense. Additionally, this paper pays attention to mistakes that result in non-existing verb forms, whereas Heyman et al. [6] only detect so-called context-dependent mistakes.

Furthermore, there was no need to automatically introduce mistakes to create an evaluation corpus, as this study had access to student-written texts with real mistakes.

Finally, the model presented in this paper is easily scalable, and does not need many computational resources, as it was able to be trained in less than a day on a Google Colab server.

5 SUMMARY

This paper focused on creating a model for detecting dt-mistakes in Dutch sentences. Sentences were preprocessed in that SpaCy's part-of-speech tagger identified the verbs. One verb per sentence received a masked ending (the dummy symbol #). A neural network-based model, based on the ULMFiT transfer learning technique, was trained to classify the sentences as *-d*, *-t* or *-dt*, providing the verb ending. If the ending did not match the ending written by the author of the sentence, the system signaled that a dt-mistake was made. The final model needed one day of training on Google Colab's servers, with a training time of around 20 minutes per epoch, which is a major benefit. This model achieved high scores on various test data sets, comparable with the results of Heyman et al. [6], although it is difficult to accurately compare the two approaches.

In the future, more inverted sentences (mainly questions in the second person singular) could be added to the training data to improve performance predicting the second person singular of verbs in inverted sentences. Furthermore, the reasons causing the failure of the BERT model for this task can be researched. Finally, part-of-speech taggers other than spaCy may further improve the current model.

If the preprocessing is adjusted to a certain extent, the model should be usable, as a practical implication, for a wide range of other correction tasks as well, such as the *die/dat* problem (mentioned above) or the difference between the homophonous diphthongs *ei* and *ij*.

The model is thus of great interest for education purposes, as it can form part of a more extensive spellchecker.

6 APPENDIX: DATA OVERVIEW

Data set		**Verb Endings**				
Name	**Subset**	**d**	**t**	**dt**	**/**	**Total**
EUR_100K	train	34,088	61,332	11,809	253,324	360,553
	valid	8,704	15,394	3,029	63,011	90,138
	total	42,792	76,726	14,838	316,335	450,691
EUR_100K_NO_/	train	34,088	61,332	11,809	0	107,229
	valid	8,704	15,394	3,029	0	27,127
	total	42,792	76,726	14,838	0	134,356
EUR_100K_EQUAL	train	11,809	11,809	11,809	0	35,427
	valid	3,029	3,029	3,029	0	9,087
	total	14,838	14,838	14,838	0	44,514
EUR_FULL	train	512,756	925,976	178,211	3,837,637	5,454,580
	valid	128,061	231,347	44,442	959,794	1,363,644
	total	640,817	1,157,323	222,653	4,797,431	6,818,224
EUR_FULL_NO_/	train	512,756	925,976	178,211	0	1,616,943
	valid	128,061	231,347	44,442	0	403,850
	total	640,817	1,157,323	222,653	0	2,020,793
EUR_FULL_EQUAL	train	178,211	178,211	178,211	0	534,633
	valid	44,442	44,442	44,442	0	133,326
	total	222,653	222,653	222,653	0	667,959

DPC_FULL	train	116,069	201,673	37,318	706,500	1,061,560
	valid	28,519	50,281	9,332	177,257	265,389
	total	144,588	251,954	46,650	883,757	1,326,949
DPC_FULL_NO_/	train	116,069	201,673	37,318	0	355,060
	valid	28,519	50,281	9,332	0	88,132
	total	144,588	251,954	46,650	0	443,192
DPC_FULL_EQUAL	train	37,232	37,232	37,232	0	111,696
	valid	9,418	9,418	9,418	0	28,254
	total	46,650	46,650	46,650	0	139,950
WIKI	train	132,530	113,211	21,581	501,889	769,211
	valid	32,878	28,537	5,351	125,536	192,302
	total	165,408	141,748	26,932	627,425	961,513
WIKI_NO_/	train	132,530	113,211	21,581	0	267,322
	valid	32,878	28,537	5,351	0	66,766
	total	165,408	141,748	26,932	0	334,088
WIKI_EQUAL	train	21,581	21,581	21,581	0	64,743
	valid	5,351	5,351	5,351	0	16,053
	total	26,932	26,932	26,932	0	80,796
WIKI_NO_/	train	169,762	150,433	58,813	0	379,018
+ DPC_FULL_EQ.	valid	42,296	37,955	14,769	0	95,020
	total	212,058	188,398	73,582	0	474,038
WIKI_NO_/	train	310,741	291,422	199,792	0	801,955
+ EURO_FULL_EQ.	valid	77,320	72,979	49,793	0	200,092
	total	388,061	364,401	249,585	0	1,002,047

TABLE 5: Overview of the used training data sets. 'EQUAL' denotes that the amounts of *-d*, *-t* are *-dt* are made equal, 'NO_/' indicates the removal of irrelevant verb endings, which were displayed with a / sign.

7 REFERENCES

[1] L. Salifou, and H. Â Naroua. (2014, Jun.). “Design of A Spell Corrector For Hausa Language.” *International Journal of Computational Linguistics*. [On-line]. 5(2), pp. 14-26. Available: https://www.cscjournals.org/library/manuscriptinfo.php?mc=IJCL-56 [May 5, 2021].

[2] G. Alafang Malema, N. Motlogelwa, B. Okgetheng and O. Mogotlhwane. (2016, Aug.). “Setswana Verb Analyzer and Generator.” *International Journal of Computational Linguistics*. [On-line]. 7(1), pp. 1-11. Available: https://www.cscjournals.org/library/manuscriptinfo.php?mc=IJCL-73 [May 5, 2021].

[3] J.S. Sumamo, and S. Teferra. (2018, Oct.). “Designing A Rule Based Stemming Algorithm for Kambaata Language Text.” *International Journal of Computational Linguistics*. [On-line]. 9(2), pp. 41-54. Available: https://www.cscjournals.org/library/manuscriptinfo.php?mc=IJCL-73 [May 5, 2021].

[4] Z. Liu and Y. Liu. (2016). “Exploiting Unlabeled Data for Neural Grammatical Error Detection.” *arXiv.org*. [On-line]. Available: http://search.proquest.com/docview/2080422559/ [Mar. 21, 2021].

[5] Y. Li, A. Anastasopoulos, and A. W. Black. (2020, Jan.). “Towards Minimal Supervision BERT-based Grammar Error Correction.” *ArXiv200103521*. [On-line]. Available: http://arxiv.org/abs/2001.03521 [Mar. 21, 2021].

[6] G. Heyman, I. Vulić, Y. Laevaert, and M.-F. Moens. (2018, Dec.). “Automatic detection and correction of context-dependent dt-mistakes using neural networks.” *Comput. Linguist. Neth. J.* [On-line]. 8, pp. 49–65. Available: https://clinjournal.org/clinj/article/view/79 [Mar. 21, 2021].

[7] L. Allein, A. Leeuwenberg, and M.-F. Moens. (2020). "Binary and Multitask Classification Model for Dutch Anaphora Resolution: Die/Dat Prediction." *ArXiv*. [On-line]. Available: https://arxiv.org/abs/2001.02943 [Mar. 21, 2021].

[8] C. Leacock, M. Chodorow, M. Gamon, and J. Tetreault. (2014). "Automated Grammatical Error Detection for Language Learners". (2nd ed). [On-line]. Available: https://www.morganclaypool.com/doi/abs/10.2200/S00562ED1V01Y201401HLT025 [Mar. 21, 2021].

[9] T. Brants and A. Franz. (2006). "Web 1T 5-gram Version 1 - Linguistic Data Consortium." 2006. [On-line]. Available: https://catalog.ldc.upenn.edu/LDC2006T13 [Mar. 21, 2021].

[10] J. Zhang, Y. Zeng, and B. Starly. (2021, Mar.). "Recurrent neural networks with long term temporal dependencies in machine tool wear diagnosis and prognosis." *SN Appl. Sci.* [On-line]. 3(4), p. 442. Available: https://link.springer.com/article/10.1007/s42452-021-04427-5 [Apr. 28, 2021]

[11] N. Verhaert and D. Sandra. (2016). "Homofoondominantie veroorzaakt dt-fouten tijdens het spellen en maakt er ons blind voor tijdens het lezen." *Levende Talen Tijdschr*. [On-line]. Available: https://lt-tijdschriften.nl/ojs/index.php/ltt/article/view/1632 [Mar. 21, 2021].

[12] "d / dt / t." Internet: https://www.vlaanderen.be/taaladvies/d-dt-t, 2021 [Apr. 28, 2021].

[13] H. Schmid. (1997). "Probabilistic Part-of-Speech Tagging Using Decision Trees," *New Methods in Language Processing*.[On-line]. pp. 154–164. Available: https://www.cis.uni-muenchen.de/~schmid/tools/TreeTagger/data/tree-tagger1.pdf [Mar. 21, 2021].

[14] M. Honnibal and I. Montani. (2017). "spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing." [On-line]. Available: https://sentometrics-research.com/publication/72/ [Mar. 21, 2021].

[15] P. Koehn. (2005). "Europarl: A Parallel Corpus for Statistical Machine Translation." *Conference Proceedings: the tenth Machine Translation Summit*. [On-line]. pp. 79–86. Available: http://mt-archive.info/MTS-2005-Koehn.pdf [Mar. 21, 2021].

[16] H. Paulussen, L. Macken, W. Vandeweghe, and P. Desmet. (2013). "Dutch Parallel Corpus: A Balanced Parallel Corpus for Dutch-English and Dutch-French." [On-line]. pp. 185–199. Available: https://link.springer.com/chapter/10.1007/978-3-642-30910-6_11 [Mar. 21, 2021].

[17] "Index of /nlwiki/." Internet: https://dumps.wikimedia.org/nlwiki/, 2021 [Apr. 28, 2021].

[18] "LIIR – Home." Internet: http://liir.cs.kuleuven.be/software_pages/dt_correction_dataset_preprocessing.php, 2018 [Mar. 21, 2021].

[19] "Circus Maximus." Internet: https://www.scholieren.com/verslag/werkstuk-geschiedenis-circus-maximus, 2007 [Mar. 21, 2021].

[20] "De gevolgen van de ontdekkingsreizen." Internet: https://www.scholieren.com/verslag/werkstuk-geschiedenis-de-gevolgen-van-de-ontdekkingsreizen, 2003 [Mar. 21, 2021].

[21] "Aquaducten." Internet: https://www.scholieren.com/verslag/werkstuk-latijn-aquaducten, 2021 [Mar. 21, 2021].

[22] "Internationale politiek België." Internet: https://www.scholieren.com/verslag/opdracht-geschiedenis-internationale-politiek-belgie, 2004 [Mar. 21, 2021].

[23] "Cold Skin." Internet: https://www.scholieren.com/verslag/boekverslag-engels-cold-skin-door-steven-herrick, 2010 [Mar. 21, 2021].

[24] J. Howard and S. Ruder. (2018). "Universal Language Model Fine-tuning for Text Classification." [On-line]. Available: http://arxiv.org/abs/1801.06146 [Mar. 21, 2021].

[25] B. van der Burgh. "110k Dutch Book Reviews Dataset for Sentiment Analysis." Internet: https://github.com/benjaminvdb/DBRD, 2019 [Mar. 21, 2021].

[26] "torch.nn - PyTorch 1.5.0 documentation." Internet: https://pytorch.org/docs/stable/nn.html [Mar. 21, 2021].

[27] S. Faltl, M. Schimpke, and C. Hackober. "ULMFiT: State-of-the-Art in Text Analysis", Internet: https://humboldt-wi.github.io/blog/research/information_systems_1819/group4_ulmfit/, 2019 [Mar. 21, 2021].

[28] "About - fast.ai," Internet: https://www.fast.ai/about/, 2020 [Mar. 21, 2021].

[29] J. Howard and S. Gugger. (2020, Feb.). "Fastai: A Layered API for Deep Learning." *Information.* 11(2). p. 108. Available: https://www.mdpi.com/2078-2489/11/2/108 [May 4, 2021].

Evidence Data Preprocessing for Forensic and Legal Analytics

Sundar Krishnan *skrishnanl@shsu.edu*
Department of Computer Science
Sam Houston State University
Huntsville, TX, USA

Narasimha Shashidhar *karpoor@shsu.edu*
Department of Computer Science
Sam Houston State University
Huntsville, TX, USA

Cihan Varol *cvarol@shsu.edu*
Department of Computer Science
Sam Houston State University
Huntsville, TX, USA

ABM Rezbaul Islam *ari014@shsu.edu*
Department of Computer Science
Sam Houston State University
Huntsville, TX, USA

Abstract

Electronic evidential data pertaining to a legal case, or a digital forensic investigation can be enormous given the extensive electronic data generation mechanisms of companies and users coupled with cheap storage alternatives. Working with such volumes of data can be tasking, sometimes requiring matured analytical processes and a degree of automation. Once electronic data is collected post eDiscovery hold or post forensic acquisition, it can be framed into datasets for analytical research. This paper focuses on data preprocessing of such evidentiary datasets outlining best practices and potential pitfalls prior to undertaking analytical experiments.

Keywords: eDiscovery, Electronic Stored Information, Digital Evidence, Digital Forensics, Digital Forensic Analytics, Legal Analytics, Machine Learning, Preprocessing, Natural Language Processing.

1. INTRODUCTION

Computers, mobile devices, smartphones, medical devices, the Internet Of Things and other electronic devices maybe used for committing crime, making law enforcement to leverage digital forensics to fight crime. These devices store, receive and transmit data that can be of critical evidential value to an investigation or legal arguments [1]. Digital evidence is now used to prosecute both civil and criminal cases when the evidence pile involves electronic devices. Ultimately, digital evidence for a case should be admissible in court and its significance explained to a jury. Digital forensic experts assist counsel through legally reliable methods to ensure Digital Evidence’s admissibility in both civil and criminal cases. Unfortunately, the volume of digital evidence can be overwhelming for digital forensic experts given the growth of technology [2]. A potential solution can be the use of analytics such as machine learning and artificial intelligence to assist in the review of evidence. After the forensic acquisition of evidence from an electronic device, digital forensic experts can export a read-only copy of raw evidence data from forensic tools to conduct analytical experiments.

In the legal system, discovery is the process that governs the right to obtain and the obligation to produce non-privileged matter relevant to any party’s claims or defenses. eDiscovery is the discovery process applied to Electronically Stored Information (ESI) or case data, such as emails, digital data from the Internet, computer files, databases, etc. The growing emphasis on paperless files and collaborative computer systems coupled with connectivity to the Internet and cheap Cloud storage has created even greater volumes of electronic information. A legal case ESI can be voluminous and can be challenging especially during the review stages [3]. This means that attorneys and other legal professionals will have to deploy and learn new technology to quicken the pace of eDiscovery while maintaining the quality of work. Legal professionals can use the Electronic Discovery Reference Model (EDRM) [4] as a starting point and is widely considered as the definitive framework for the eDiscovery process.

Many eDiscovery solutions/tools have focused on improving collection efficiency and reducing data review effort for long. Legal analytics is the management process of extracting actionable knowledge from data to assist in-house legal leaders and decision-makers [5]. Few use cases for legal analytics involve eDiscovery efficiency, motion forecasting, process improvement, legal strategy, comparative legal costs, billing optimization, settlement award, resource management, and financial operations. To summarize, legal analytics tools help lawyers make data-driven decisions on which to build their legal strategies [6]. During eDiscovery, once relevant case data is collected and loaded into a storage platform, legal teams begin reviewing the data. Data reviews can be time-consuming and are the bulk of litigation costs. Thus, leveraging technology through an established framework can greatly help with speed and accuracy during reviews. In 2012, EDRM proposed Technology-Assisted Review (TAR) [7] and has since been steadily gaining popularity with the industry as an essential tool in eDiscovery. The TAR framework (also known as predictive coding) refers to a document review approach in eDiscovery that leverages computer algorithms to identify and tag potential documents based on keywords and metadata. TAR in it’s original form is a multi-step process spanning anywhere between 6 to 10 steps depending on whether Simple Active Learning (SAL) or Simple Passive Learning (SPL) was being used. The second generation of Technology-Assisted Review (TAR 2.0), however, consists of Continuous Active Learning (CAL), which enables a system to continuously analyze the machine learning results (in the background) as humans review documents without the need to begin by analyzing static, randomized samples [8]. The result is a non-iterative and continuously improving implementation of TAR as the review progresses by re-ranking the entire data set with each new batch of data [8]. Continuous Active Learning indicates that the system uses the updated model to continuously promote case documents to the top of the review queue that has the highest probability of being responsive to the case [9]. Thus, TAR 2.0 has many advantages over TAR 1.0. In TAR 1.0, experts do the initial training, and it is less effective because it cannot learn from subsequent decisions. TAR 1.0 also cannot handle rolling productions without having to start over [10]. In TAR 2.0, all human review decisions automatically train and update the system predictions as new human classifications are made.

Technology Assisted Review (TAR) has established itself into standard e-discovery practices with a key benefit of expediting the document review process. TAR has also garnered favor with judges familiar with its benefits [11] and also has judges refusing to compel parties to apply TAR [12]. Analytical techniques such as Machine Learning (ML) (supervised or un-supervised), Artificial Intelligence (AI), Deep Learning, Neural Networks, Statistical approaches, etc. fall under TAR technology umbrella. Since recently, these techniques have gained popularity with legal firms and eDiscovery solutions vendors with a goal to expedite the organization and prioritization of document collection and minimize review efforts. These techniques help save costs and reduce time in helping to identify relevant data. ML-based solutions such as Brainspace [13] can perform conceptual clustering by reading case documents, searching for relevant words, and clustering them into groups based on their contents [14]. AI is a very useful assistant when helping to identify relevant data by leveraging supervised learning. However, these techniques have limitations as they clearly do not run the investigation but, merely assist in speeding up the overall process.

Data analytics is a broad term that refers to the use of various techniques that find meaningful patterns, predict the future, and give insights into data. Data analytics is not new to digital forensics or to the legal world and can be as simple as employing statistics in decision making. Few enabling fields of data analytics are data science and data engineering. Data science is a process of testing, evaluating, and experimenting to create and apply new data analytic techniques. Data engineering makes data useful by helping structure data making it easier for application and human consumption. Data analytics has greatly manifested in the last few years as we focus more on business intelligence and the real-time analysis of data [15]. The explosion of smartphone usage, coupled with easier Internet connectivity and low costs of Cloud storage, has converted data analytics into a buzzword. The need to derive meaningful insights into customer or business data has pushed disciplines such as text/image mining, predictive modeling, etc. The technical aspects of analytics can be found in the emerging fields of machine learning techniques such as neural networks, decision trees, logistic regression, linear/multiple regression analysis and classification. All of these disciplines require clean raw data for input and the process of cleaning/transforming raw data is known as preprocessing. Often raw data is likely to be imperfect, noisy, inconsistent, and sometimes redundant, making it unfit for analysis. Analytical experiments greatly depend on the quality of input data, and as such, results can be skewed or incorrect if data was not preprocessed correctly prior to applying algorithms. These days, law firms, eDiscovery vendors, legal and forensic researchers, have all started to venture into experimenting with advanced analytical techniques such as Machine Learning and Artificial Intelligence. Legal and forensic analytical experiments can be around actual digital forensic investigations of a case, reviews of case ESI during eDiscovery, staged experiments for research and process optimization. There have been promising results when applying these advanced analytical techniques in legal eDiscovery yielding in direct financial advantages. However, there exists caution in the legal industry and digital forensics investigations when leveraging such techniques as applying analytics is still in a nascent stage with courts being the ultimate proving ground in validating their use. In this article, the authors share their best-practices when preparing for such advanced analytical experiments in a legal setting under TAR or within a digital forensic investigation scope. Suggested best practices are guidelines that can lower risk and improve the statistical model's efficiency and accuracy when employing analytical techniques such as Machine Learning or Artificial Intelligence to work in a legal setting or forensic investigation.

2. RELATED WORK

Data preprocessing and dimensionality reduction is an integral step in any analytical experiment leveraging statistics, machine learning, artificial intelligence, neural networks, etc. The number of features, quality of input data, and the useful information that can be derived from it directly impacts the ability of the algorithms and eventually the result. A typical use case in analytic experiments in eDiscovery is around reviewing emails within the case ESI. Email preprocessing can help identify spam, categorize emails and mitigate phishing attacks. Ruskanda [16] studied the effect of preprocessing of emails on spam email detection techniques using supervised spam classifier algorithms: Na¨ıve Bayes and Support Vector Machine. Kumara et al. [17] propose an enhanced data preprocessing approach for multi-category email classification by ignoring the signatures on emails, special characters, and unwanted words. Their proposed model was evaluated using various classifiers and showed that the proposed data preprocessing to email classification is superior to the existing approach. Emails can be complex to parse and process due to branching, forwarding, attachments, multiple languages, signatures, footers, disclaimers, auto-generated phishing warnings, URLs, etc. Tang et al. [18] in a cascaded approach, propose leveraging Support Vector Machines (SVM) to clean up emails by addressing non-text filtering, paragraph normalization, sentence normalization, and word normalization. Emails branch and can sometimes render a partial picture of the whole conversation. A single longest thread alone can only track a linear, back-to-back conversation. Instead, to factor the whole conversation, branching emails should be considered and grouped [19]. Another focus area of legal analytics during reviews is data from social media that can be littered with words from multiple languages, jargon, code words, abbreviations, shortened words, etc. Uysal et al. [20] examine the impact of preprocessing on text classification using benchmark datasets. They concluded that the choice and combinations of preprocessing tasks may provide a significant improvement on classification accuracy depending on the domain and language. Kantepe et al. [21] propose a preprocessing framework for Twitter bot detection with reasonable accuracy using a machine learning supervised classification approach. Etaiwi et al. [22] investigate the effects of preprocessing steps on the accuracy of reviews spam detection by applying machine-learning algorithms against a labeled dataset of hotel reviews. Data from social media can be complex simply due to multiple languages used, sharing, liking, commenting, etc. There exists a gap in literature focusing on preprocessing challenges and best-practices when working on analytical experiments or research with forensic evidence and legal case data. While existing literature focusses on generalized approaches towards various data preprocessing techniques, algorithms, etc. there is little contribution towards applying such methodology towards industry specific use-cases. In this paper, the authors discuss best-practices and potential issues for legal and forensic data analysts during data preprocessing when working in forensic and legal investigations or analytical tasks.

3. DATA PREPROCESSING FOR FORENSIC AND LEGAL ANALYTICS

A caseload of digital evidence can be viewed as a data-lake that can translate into meaningful datasets for analytical experiments. To understand the depth of analytical algorithms, the features (attributes or variables) in the evidence/case data, and what they represent are to be well understood. This section delves into best practices when preparing for analytical experiments using evidentiary case data during legal analytics or forensic investigations.

3.1 Research Methodology

The methodology of this paper includes reviewing existing literature, examining best-practices and potential pitfalls during data preprocessing in forensic and legal investigations in addition to following current industry trends.

3.2 Identify Analytical Aim/Problem/Objective

Like any analytical experiments, legal and forensic analytics will need to identify aims to accomplish or problems to be solved prior to the start of experiments. They can help devise a strategy and identify the data that needs to be collected. Aims or problems are usually derived from the investigation scope, forensic protocol, or legal case scope. In a legal case, scope can be defined as the extent of ESI discovery that the parties agree to produce for the case and is generally defined by the Federal Rule of Civil Procedure 26(b)(1) [23]. During a digital forensic investigation, the scope and forensic protocol can be obtained from the investigation plan, security incident response or warrants. Scope limitations may be in effect due to time availability, forensic skills availability, forensic tool availability, budget, privacy or opposing interests. Figure 1 highlights the sources for deriving Aim/Problem/Objective in legal and forensic analytics.

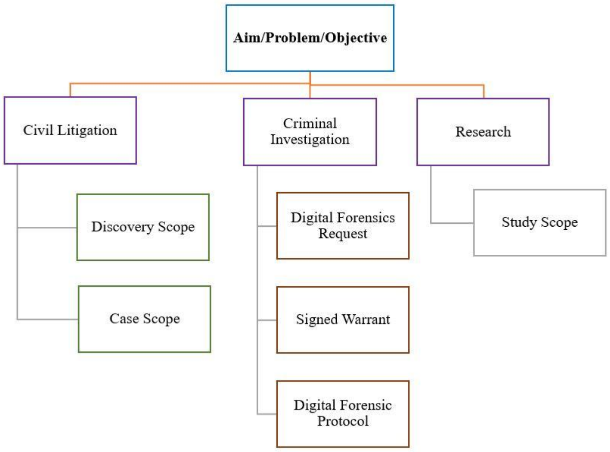
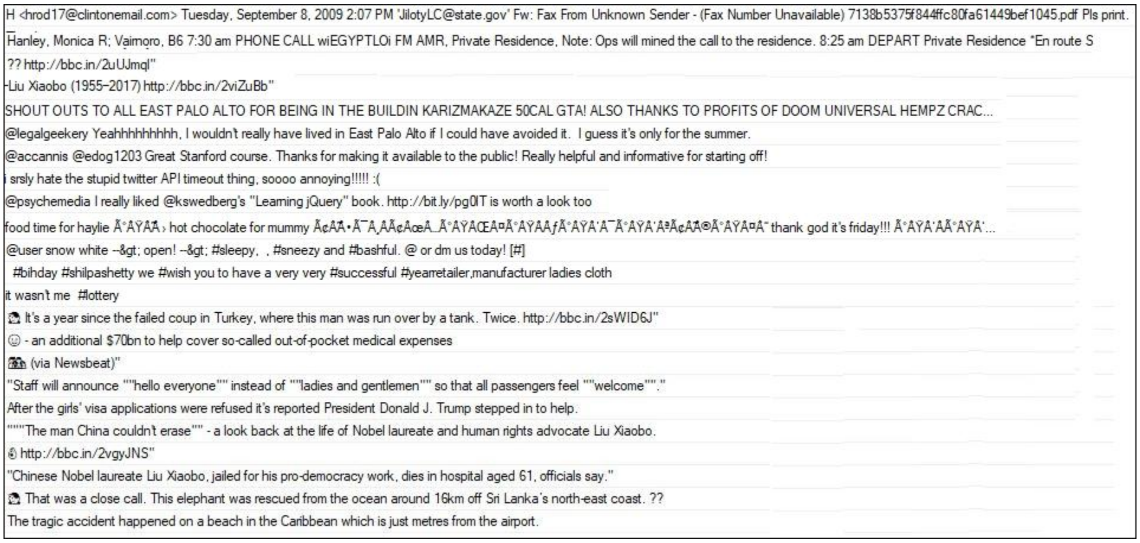


FIGURE 1: Sources of analytical aim/problem/objective in legal and forensic analytics.

3.3 Understanding Case or Evidence Data

To gain actionable insights into a legal case or forensic investigation, the appropriate data from case ESI or evidence must be sourced and cleansed prior to conducting analytical experiments. Care must be taken not to spoil the data by hampering its integrity, and, thus a true, verifiable copy of the data may be used for analytical experiments. There are two key stages of data Understanding: Assessment and Exploration. The first step is assessment during which, availability, format, storage, source, features, relevance, quality, reliability, etc., are explored. During the exploration step, missing values, outliers, bias, balance, etc., are explored. Case ESI data or evidence data post forensic acquisition can arrive from various devices/sources and in different raw formats. Data can be uploaded into a database or into spreadsheets for easy exploration. Statistical formulae can be used to further explore balance, mean, variance, etc. Feature engineering can then help normalize and scale data.

Few types of analytics that are having a significant impact on eDiscovery and forensic investigations are Machine Learning (ML), Convolutional Neural Networks (CNN), and Natural Language Processing (NLP). Machine learning uses mathematical models to assess enormous datasets, make predictions and learn from feedback. NLP allows machines to "understand" natural human language, thereby enabling computers to effectively communicate in the same language as their users. Although NLP and its sister study, Natural Language Understanding (NLU) are constantly advancing in their ability to compute words and text, human language can be complex, ever-evolving, fluid, and inconsistent thereby presenting serious challenges that NLP is yet to completely overcome. Since case data can mostly comprise of text, NLP is a suitable technique that is commonly used. Table I outlines few challenges when working with text-based case data. Figure 2 shows potential issues with raw data of a legal case ESI.



```
H <hrod17@clintonemail.com> Tuesday, September 8, 2009 2:07 PM 'JilotyLC@state.gov' Fw: Fax From Unknown Sender - (Fax Number Unavailable) 7138b5375f844ffc80fa61449bef1045.pdf Pls print.
Hanley, Monica R; Vaimoro, B6 7:30 am PHONE CALL wiEGYPTLOi FM AMR, Private Residence, Note: Ops will mined the call to the residence. 8:25 am DEPART Private Residence "En route S
?? http://bbc.in/2uUJmql"
Liu Xiaobo (1955–2017) http://bbc.in/2viZuBb"
SHOUT OUTS TO ALL EAST PALO ALTO FOR BEING IN THE BUILDIN KARIZMAKAZE 50CAL GTA! ALSO THANKS TO PROFITS OF DOOM UNIVERSAL HEMPZ CRAC...
@legalgeekery Yeahhhhhhhhh, I wouldn't really have lived in East Palo Alto if I could have avoided it.  I guess it's only for the summer.
@accannis @edog1203 Great Stanford course. Thanks for making it available to the public! Really helpful and informative for starting off!
i srsly hate the stupid twitter API timeout thing, soooo annoying!!!!! :(
@psychemedia I really liked @kswedberg's "Learning jQuery" book. http://bit.ly/pg0IT is worth a look too
food time for haylie Ã°ÅŸÃÅ› hot chocolate for mummy Ã¢Ã€•Ã¯Â¸ÂÃ¢Âœ Â_Ã°ÅŸÂŒÃ¤Ã°ÅŸÃƒÃ°ÅŸÂ'Ã¯Ã°ÅŸÂ'ÂªÃ¢Ã€®Ã°ÅŸÂ¤Ã¨ thank god it's friday!!! Ã°ÅŸÂ'ÃÃ°ÅŸÂ'...
@user snow white --&gt; open! --&gt; #sleepy,  , #sneezy and #bashful. @ or dm us today! [#]
 #bihday #shilpashetty we #wish you to have a very very #successful #yearretailer,manufacturer ladies cloth
it wasn't me  #lottery
It's a year since the failed coup in Turkey, where this man was run over by a tank. Twice. http://bbc.in/2sWID6J"
☺ - an additional $70bn to help cover so-called out-of-pocket medical expenses
(via Newsbeat)"
"Staff will announce ""hello everyone"" instead of ""ladies and gentlemen"" so that all passengers feel ""welcome""."
After the girls' visa applications were refused it's reported President Donald J. Trump stepped in to help.
"""The man China couldn't erase"" - a look back at the life of Nobel laureate and human rights advocate Liu Xiaobo.
http://bbc.in/2vgyJNS"
"Chinese Nobel laureate Liu Xiaobo, jailed for his pro-democracy work, dies in hospital aged 61, officials say."
That was a close call. This elephant was rescued from the ocean around 16km off Sri Lanka's north-east coast. ??
The tragic accident happened on a beach in the Caribbean which is just metres from the airport.
```

FIGURE 2: Sample raw text in a case ESI or digital forensic evidence prior to preprocessing. Contains garbled characters, Unicode, email addresses, shorthand, slang, URLs, emoji and hashtags.

The use of programming languages, software, and automation technology can sometimes impact data integrity. Storage of raw case/evidence data on databases should be undertaken with caution to support Unicode, logos, signatures, image & video pixel resolution, gifs, VR media, etc. Database or file-system transactions should not alter the state of raw evidence data. For example, for processing Facebook data in Arabic or French language containing emoji (a true-copy from a case ESI or digital evidence) stored on a SQL Server database instance, should consider the schema (column-level) design for Unicode and multilingual language support. Similarly, transacting with this database using Python programming language to perform analytical research should be undertaken with caution as read writes into the database can accidentally ignore/suppress Unicode support, thereby impacting data integrity and experiment results. Thus, a cursory glance at raw data should be undertaken before identifying and designing technology platforms for analytics.

3.3 Technology Selection

Digital Forensic tools, email processing tools, social media crawlers, eDiscovery solutions, and various other extraction/parsing tools are some of the technology-driven tools that can help extract and export data from case evidence. Not all tools export extracted data in the same format. Thus, for analytical experiments, data has to be collated into a single dataset with necessary features. Appropriate computer programs can be leveraged to legally obtain social media website data via their defined application programming interfaces (API). Relational databases can be used to collect and store data following which queries may be used to create datasets. Randomly, exported data from the tool will need to be validated against reported/observed evidence (device) data for tool accuracy and dependability. The assistance of data scientists, data engineers, statisticians, domain experts and Information Technology staff may be required when conducting any legal analytical experiments.

3.4 Digital Forensics

There exists an interplay between eDiscovery and digital forensics [24] when data from evidence will need to be forensically extracted for legal arguments and investigation. The collection phase of eDiscovery is when digital forensic professionals are often engaged to protect data integrity and to bring forth the data stored on digital evidence. Digital forensic tools export evidence data into various formats. Note that not all forensically acquired data (evidence) may be directly ready for analytical experiments. Images, audio, and video files may contain hidden data or be deep-fake needing to be suitably addressed. Few variations of legal analytical research may involve forensic investigations. For example, predicting friends using social media data or clustering documents related to a crime. During such research, the investigative skills of digital forensic professionals may be leveraged to validate results.

3.5 Identify Key Features

In a legal case-load of evidence, data within the evidence device/source is not always ready for immediate analytical experiments. Case evidence data often can be found as digital files from various software programs or plainly skimmed off the Internet. This makes identification of data within such data a prerequisite, as data can be generally voluminous and uncured. Key features (attributes or variables) of data will need to be identified for the legal case. Identifying key features ahead of an analytical experiment requires planning and assistance from technical experts on the case. Key features may start from a wish-list but should be scoped to translate into being technically feasible collection while mainlining data integrity all through the process. For example, if the case arguments hinge upon presence of the client at specific locations over a time, then details such as timestamps and geographical location from data are key features that need to be collected into datasets. In another example, if the case arguments hinge upon the use of a computer for certain Internet activities, features from case-data such as login data (of both computer and online websites such as timelines, authentication tokens, the identity used), web activity (timelines, posts, likes, dislikes, and comments) and geographical location data from network traffic may be of use. Ancillary features such as online responses from friends/strangers of the defendant/client may add noise and degrade the analytical algorithms in the experiments. Multiple datasets of such key features can be then prepared for individual analytical experiments.

3.6 Data Threads

Disentangling conversations mixed into a single stream of messages can create challenges unless properly handled and carved into detached yet linked data. Further complications arise when conversations are peppered with slang, abbreviations, URLs, etc. A common occurrence of such conversations are long email threads that are often the first to be reviewed during eDiscovery following “The Longest Thread Policy” [19]. An email thread is a group of emails all originating from the same email that branch off in many directions as receivers (copied or blind-copied) forward the email to different recipients. Sometimes, other email threads can interweave into threads that can complicate a walk. Slicing emails from threads for analytical experiments can cause data loss or introduce noise. In some instances, senders may manually remove or edit certain email body when forwarding or replying. Such data loss should be monitored. Automation tools that help parse emails should be carefully chosen to report any such discrepancies. Similarly, conversations on social media platforms can branch (like a tree) into multiple senders and receivers. A conversation path must be identified to isolate actors/subjects, timelines, and their conversations. Improper handling of such lengthy strings of data can also lead to missing out on the context of the whole conversation. Parsing attachments, embedded videos or images in such threads can add to the complexity, thus requiring design considerations on datasets.

Description	Expression
Loan-words in English of foreign origin	bona fide ad nauseam, en masse, faux pas, fait accompli, modus operandi, persona non grata, quid pro quo bon voyage, pro bono, status quo, avatar, guru, chilly (means peppers in Indian language), hullabaloo, mulligatawny, Chop chop, Feng shui, Coolie, Nankeen (durable cloth in Mandarin)
Sarcasm	“Is it time for your medication or mine?” “My favorite thing to do at 5AM is to go to the Airport. How about you?” “That’s just what I needed today!”
Irony	“The fire station burned down” “The traffic cop got his license suspended because of unpaid parking tickets”
Errors in text or speech (Psycholinguistic classification like deletion, blends, addition, omission, etc. [25])	“Bake my bike” “He pulled a pantrum” “Both sicks are kids”
Colloquialisms and slang	“I’m fixin’ to go to the park” “Blimey” - exclamation of surprise, “Chockablock” - something that is completely filled, “Dodgy” - something less than safe or secure, “Lemon” - a purchase that is unreliable

TABLE 1: Common language and text limitations in case evidence data.

3.7 Data Correlation

Finding correlations in data from multiple data sources may be needed as part of analytical experiments. Correlation is like finding a pattern on wallpaper and is a statistical-based information analysis technique of analyzing relationships between two or more features (variables). For example, correlating data from sources such as company email and Facebook activity may be needed for legal arguments. In such situations, data for emails may be extracted from an exchange server or Microsoft 365 and Facebook data may be extracted from a smartphone. Creating datasets using both sources of data will need design insights and adequate planning.

3.8 Goodness of Fit

Model fitting is a measure of how well a machine learning model generalizes data that is similar to which it was trained for [26]. A good model fit is a statistical hypothesis test that of a model that accurately approximates the output when it is provided with unseen inputs. The goodness of fit of a statistical model describes how well it fits a set of observations. Over fitting a model captures the noise and outliers in the data along with the underlying pattern. Such models usually have high variance and low bias. Under fitting a model occurs when the model is unable to capture the underlying pattern of the data and is too simple. Such models usually have a low variance and a high bias. Bias and variance are key risks in analytical experiments and can be best addressed by implementing statistical best practices. Bias exists in all data driven experiments, but the question is how to identify and remove it from the experiment. Bias can skew results and might negatively impact the effectiveness of the experiment's algorithms. To avoid bias, careful planning of the experiment is needed, and a balance between transparency and performance has to be maintained. Bias in analytical experiments can eventually derail a legal case.

3.9 Data Loss

Inadvertent data conversions can lead to data loss. Care should be taken in instances when emoji, glyphs, Unicode scalars, favicons, emoticons, nicknames, slang words, abbreviations, Anglicized language, etc. are embedded in text. Encoded conversations, embedded images or videos can change the meaning to a plain text conversation but may also hold a secret meaning for the intended targets. Data transformation, filtering, encoding, removing email appends (logos, banners, system-generated phishing warnings, printer ink-friendly messages), etc. can all lead to data loss. However, this must be documented and not adversely impact the aim of the analytical experiment.

3.10 Data Leakage

Often encountered during predictive analytics, data leakage is when information from outside the training dataset is used to create a model. This can be accidental sharing of information between the test and training data during the experiment, or during data preprocessing. Data Leakage can lead to false assumptions about the performance of the analytical model. Generally, if the analytical model is too good to be true, we should be suspicious.

3.11 Sensitive Data and Privacy

Sensitive data is any data such as personally identifiable information (PII), Protected Health Information (PHI), Payment Card Industry (PCI) data, Intellectual Property (IP), and other important business data. Analytical experiments may need to use such sensitive data. Legal firms have to comply with common industry regulatory standards for data protection and privacy such as; the General Data Protection Regulation (GDPR), the California Consumer Privacy Act (CCPA), Health Insurance Portability and Accountability Act (HIPAA), the Payment Card Industry Data Security Standard (PCI DSS), standards from the International Organization for Standardization (ISO), and others. Prior identification of sensitive data by manual or by leveraging pre-tuned industry tools is recommended. Specific use approvals from data custodians or identified authority is recommended prior to starting on analytic experiments. Processing of sensitive data through encryption, tokenization, redaction, masking, or de-identification maybe needed. For example, masking of last names may be required, or certain geographical location data may need to be obfuscated to protect privacy and identity. If so, these features may need to be dropped or encoded accordingly during analytical preprocessing. If authorized to use raw data for experiments, care must be taken for storage and distribution of results lest they accidentally expose sensitive data.

4. DATA MANAGEMENT DURING ANALYTICS

A disciplinary approach should be maintained during preprocessing and filtering of data when building a dataset. Multiple copies of data or datasets stored indiscriminately on storage drives/network can increase security and privacy risks. Industry best practices should be implemented, or organization policies followed when creating copies of case data. To avoid spoliation and accidental evidence corruption, a read-only copy of original raw evidentiary data should be carefully generated prior to use in any research or experiments.

4.1 Data Integrity

Data preprocessing steps can be lengthy when arriving at the best set of features for the analytical experiments. Care should be taken on data integrity as indiscriminate processing can truncate or manipulate data. For example, careless rounding of a float dataype or encoding a string datatype into a numeric datatype can impact the performance of the model and impact experiment conclusions. When exporting data off automation or forensic tools, similar caution should be employed lest the tools accidentally convert, format, or truncate data (data types). For example, when exporting timelines from a smartphone post digital forensic investigation, care should be taken to maintain the date and time format of data, timezones especially when the device was used across countries. Transposing such data to adjust for the analytical experiment's needs should be undertaken with caution and documented.

4.2 Security and Access

Proper access (authorization and authentication) to data should be considered before the start of any analytical experiments. Access to data can be limited to read-only. Data shares with other teams should be part of authorization protocols. Similarly, reports and analysis from analytical experiments should be carefully shared with those who are authorized to receive them. Once analytical experiments are completed, authorization should be revoked to case data. Unless allowed by enterprise policy, use caution when sharing case data or analytical experiment results over emails or through enterprise messaging/chat applications. Industry best practices around security and privacy should be followed such as, implementing Data Loss Prevention (DLP) controls on endpoints and monitoring of network traffic.

4.3 Policy and Guidelines

Legal firms, eDiscovery/forensic practitioners, forensic labs, and vendors should ensure data management and governance, privacy, ethics and security policies are in place when working with case data. A separate policy and set of standards may be envisioned to address analytical research.

4.4 Backup and Retention

Plans for analytical research and experiments should follow enterprise backup and retention procedures. Pre-determined backup (storage) locations must be identified, and retention period defined.

4.5 Destruction

Upon completion or termination of analytical research and/or experiment(s) using case data, the concerned Information Technology or Security teams should be notified. Industry best practices, standards [27], [28] or enterprise defined policies may be employed for data clean-up destruction) processes to counter residual data. For example, if a Cloud based storage location or a portable storage-media were used as part of the analytical research and/or experiment(s), proper procedures must be followed to wipe the storage media or engage with the Cloud Service Provider to undertake the same. Likewise, systems used during the analytical research and/or experiment(s) should be subject to safe wiping policies and procedures.

5. CONCLUSION

Advanced analytical research and experiments are these days undertaken in-house by teams of data scientists with a background in legal, eDiscovery, Information Technology and Statistics. Forensic and legal analytics has come to the forefront of investigations and technology-assisted reviews given the recent focus in Machine Learning, Artificial Intelligence, and Deep Learning. In a legal case, digital evidence may be present as digital devices or Internet data. Extracting data off such evidence can be voluminous and can burden the review process during eDiscovery. Advanced analytical processing by digital forensic and legal professionals can come to the rescue of winnowing and interpreting large volumes of evidence data for establishing patterns, intent, and motives. Also, forensic, and legal analytical approaches can be used in forensic investigations to reduce evidence search time, gain insight into suspect's activities, clustering suspect profiles, optimize legal costs, case billing, motion prediction, legal strategizing, etc. All legal analytical research or experiments require data as inputs and raw data may not always be of the best quality for direct consumption. This paper outlines best practices and approach for preprocessing legal data prior to forensic and legal analytics. Leveraging analytics can greatly assist in manual case reviews and investigations but should not be considered as their replacement and solely relied upon as applying analytics is still considered as nascent in legal minds. It can be safely predicted that forensic and eDiscovery experts will soon need to add analytical and statistical skills to their knowledgebase to leverage them in their work and explain the significance of these fields to a jury when offering expert opinions and interpreting investigation findings. In future work, the authors propose to focus on assessing the performance of legal analytical techniques to test and confirm the accuracy of preprocessing of evidentiary case data.

6. REFERENCES

[1] “Digital Evidence and Forensics.” Internet: https://nij:ojp:gov/digital-evidence-and-forensics, [Mar. 02, 2021].

[2] D. Quick and K. K. R. Choo, Dec 2014, “Impacts of increasing volume of digital forensic data: A survey and future research challenges,” Digit. Investig., [On-line] vol. 11, no. 4, pp. 273–294,Available: https://www.sciencedirect.com/science/article/abs/pii/S1742287614001066, [Mar. 02, 2021].

[3] S. Krishnan, A. Neyaz, and N. Shashidhar, 2019, “A Survey of Security and Forensic Features In Popular eDiscovery Software Suites,”. [On-line]. Available: https://www:cscjournals:org/manuscript/Journals/IJS/Volume10/Issue2/IJS-152:pdf, [Mar. 02, 2021].

[4] Electronic Discovery Reference Model, Internet: https://edrm.net/resources/frameworks-and-standards/, [Mar. 02, 2021].

[5] “Legal Analytics.”, Internet: http://www:argopoint:com/legalanalytics, [Mar. 02, 2021].

[6] “What is Legal Analytics?”, Internet: https://www:lexisnexis:com/community/lexis-legal-advantage/b/insights/posts/what-is-legal-analytics, 2019, [Mar. 02, 2021].

[7] EDRM, “Technology Assisted Review.”, Internet: https://edrm:net/resources/frameworks-and-standards/technologyassisted-review/, [Mar. 04, 2021].

[8] S. Kernisan, “TAR 1.0 or TAR 2.0: Which method is best for you?”, Internet: https://www:casepoint:com/blog/tar-1-0-versus-tar-2-0/, [Mar. 04, 2021].

[9] G. Taranto, “The Evolution of TAR”, Internet: https://www:law:com/2020/12/31/the-evolution-of-tar/?slreturn=20210110063112, 2020, [Mar. 04, 2021].

[10] J. Kerry-Tyerman, “Why Machine Learning Matters in Ediscovery”, Internet: https://www:everlaw:com/blog/2018/01/03/machine-learning-in-ediscovery/, 2018, [Mar. 04, 2021].

[11] Casetext, “Moore v. Groupe, 868 F. Supp. 2d 137”, Available: https://casetext:com/case/moore-v-groupe, 2012, [Mar. 04, 2021].

[12] Hyles v. City of New York et al, No.1:2010cv03119 - Document 97 (S.D.N.Y. 2016), Available:https://law:justia:com/cases/federal/district-courts/new-york/nysdce/1:2010cv03119/361399/97/, 2016, [Mar. 06, 2021].

[13] Brainspace: Make Smarter, Faster, & More Informed Decisions., Available: https://www:brainspace:com/, [Mar. 07, 2021].

[14] Artificial intelligence and machine learning in e-discovery and beyond., Available: https://www2:deloitte:com/ch/en/pages/forensics/articles/AI-and-machine-learning-in-E-discovery:html, [Mar. 07, 2021].

[15] L. Wilson, “Enterprise AI: Data Analytics, Data Science and Machine Learning”, Available: https://www:cio:com/article/3342421/enterprise-ai-data-analyticsdata-science-and-machine-learning:html, 2018, [Mar. 07, 2021].

[16] F. Z. Ruskanda, Mar 2019, “Study on the Effect of Preprocessing Methods for Spam Email Detection,” Indones. J. Comput., [On-line] vol. 4, no. 1, p. 109, Available: http://www:mail-abuse:com/, [Mar. 07, 2021].

[17] B. A. Kumara, M. M. Kodabagi, T. Choudhury, and J.-S. Um, Jan 2021, “Improved email classification through enhanced data preprocessing approach,” Spat. Inf. Res., [On-line] pp. 1–9, Available: https://link:springer:com/article/10:1007/s41324-020-00378-y, [Mar. 07, 2021].

[18] J. Tang, H. Li, Y. Cao, and Z. Tang, 2005, “Email data cleaning,” in Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min. New York, New York, USA: ACM Press, [On-line] pp. 489–498, Available: http://portal:acm:org/citation:cfm?doid=1081870:1081926, [Mar. 07, 2021].

[19] J. Greer, “Email Threading in eDiscovery: The Longest Thread Policy,” Internet: https://www:digitalwarroom:com/blog/emailthreading-ediscovery-problems-with-longest-thread, 2019, [Mar. 08, 2021].

[20] A. K. Uysal and S. Gunal, Jan 2014, “The impact of preprocessing on text classification,” Inf. Process. Management., [On-line] vol. 50, no. 1, pp. 104–112, Available: https://doi.org/10.1016/j.ipm.2013.08.006., [Mar. 08, 2021].

[21] M. Kantepe and M. C. Ga˜niz, Oct 2017, “Preprocessing framework for Twitter bot detection” in 2nd Int. Conf. Comput. Sci. Eng. UBMK 2017. Institute of Electrical and Electronics Engineers Inc., [On-line] pp. 630–634, Available: https://doi.org/10.1109/UBMK.2017.8093483, [Mar. 12, 2021].

[22] W. Etaiwi and G. Naymat, Jan 2017, “The Impact of applying Different Preprocessing Steps on Review Spam Detection,” in Procedia Computer Science., vol. 113. Elsevier B.V., [On-line] pp. 273–279., Available: https://doi.org/10.1016/j.procs.2017.08.368., [Mar. 12, 2021].

[23] Rule 26. Duty to Disclose; General Provisions Governing Discovery— Federal Rules of Civil Procedure — US Law — LII / Legal Information Institute., Internet: https://www:law:cornell:edu/rules/frcp/rule 26#rule 26 a 1 B, [Mar. 10, 2021].

[24] S. Krishnan and N. Shashidhar, Mar 2021, “Interplay of Digital Forensics in eDiscovery,” IJCSS, [On-line] vol. 15, issue 2, pp 19-44, Available: https://www.cscjournals.org/manuscript/Journals/IJCSS/Volume15/Issue2/IJCSS-1602.pdf, [Mar. 19, 2021].

[25] “Speech error - Wikipedia.”, Internet: https://en:wikipedia:org/wiki/Speech error, [Mar. 22, 2021].

[26] Definition: Model fitting, Internet: https://www:educative:io/edpresso/definition-model-fitting, [Mar. 25, 2021].

[27] NIST, “Guidelines for Media Sanitization, Special Publication 800-88”, Internet: https://nvlpubs:nist:gov/nistpubs/SpecialPublications/NIST:SP:800-88r1:pdf, 2014, [Mar. 29, 2021].

[28] N. I. S. Program, “DoD 5220.22-M, Operating Manual”, Internet: https://www:esd:whs:mil/Portals/54/Documents/DD/issuances/dodm/522022M:pdf, 2006, [Mar. 29, 2021].

INSTRUCTIONS TO CONTRIBUTORS

Computational linguistics is an interdisciplinary field dealing with the statistical and/or rule-based modeling of natural language from a computational perspective. Today, computational language acquisition stands as one of the most fundamental, beguiling, and surprisingly open questions for computer science. With the aims to provide a scientific forum where computer scientists, experts in artificial intelligence, mathematicians, logicians, cognitive scientists, cognitive psychologists, psycholinguists, anthropologists and neuroscientists can present research studies, International Journal of Computational Linguistics (IJCL) publish papers that describe state of the art techniques, scientific research studies and results in computational linguistics in general but on theoretical linguistics, psycholinguistics, natural language processing, grammatical inference, machine learning and cognitive science computational models of linguistic theorizing: standard and enriched context free models, principles and parameters models, optimality theory and researchers working within the minimalist program, and other approaches. IJCL is a peer review journal and a bi-monthly journal.

To build its International reputation, we are disseminating the publication information through Google Books, Google Scholar, J Gate, ScientificCommons, Docstoc and many more. Our International Editors are working on establishing ISI listing and a good impact factor for IJCL.

The initial efforts helped to shape the editorial policy and to sharpen the focus of the journal. Started with Volume 12, 2021, IJCL aims to appear with more focused issues related to computational linguistics studies. Besides normal publications, IJCL intend to organized special issues on more focused topics. Each special issue will have a designated editor (editors) – either member of the editorial board or another recognized specialist in the respective field.

We are open to contributions, proposals for any topic as well as for editors and reviewers. We understand that it is through the effort of volunteers that CSC Journals continues to grow and flourish.

IJCL List of Topics:
The realm of International Journal of Computational Linguistics (IJCL) extends, but not limited, to the following:

- Computational Linguistics
- Computational Models
- Computational Theories
- Corpus Linguistics
- Formal Linguistics-Theoretic and Grammar Induction
- Information Retrieval and Extraction
- Language Generation
- Language Learning
- Linguistics Modeling Techniques
- Linguistics Theories
- Machine Translation
- Models of Language Change and its Effect on Lingui
- Models that Address the Acquisition of Word-order
- Models that Combine Linguistics Parsing
- Models that Employ Statistical/probabilistic Gramm
- Models that Employ Techniques from machine learning
- Natural Language Processing
- Quantitative Linguistics
- Speech Analysis/Synthesis
- Speech Recognition/Understanding
- Spoken Dialog Systems
- Web Information

CALL FOR PAPERS

Volume: 12 - **Issue:** 3

i. Paper Submission: June 30, 2021 **ii. Author Notification:** July 31, 2021

iii. Issue Publication: August 2021

CONTACT INFORMATION

Computer Science Journals Sdn BhD
B-5-8 Plaza Mont Kiara, Mont Kiara
50480, Kuala Lumpur, MALAYSIA

Phone: 006 03 6204 5627
Fax: 006 03 6204 5628
Email: cscpress@cscjournals.org

CSC PUBLISHERS © 2021
COMPUTER SCIENCE JOURNALS SDN BHD
B-5-8 PLAZA MONT KIARA
MONT KIARA
50480, KUALA LUMPUR
MALAYSIA

PHONE: 006 03 6204 5627
FAX: 006 03 6204 5628
EMAIL: cscpress@cscjournals.org