Master's thesis
Master's Programme in Data Science

Discourse Act Classification in Asynchronous Online Forum Discussions

Rick Joosten

September 21, 2020

Supervisor(s): Antti Ukkonen

Examiner(s): Prof. Indrė Žliobaitė
Antti Ukkonen

University of Helsinki
Faculty of Science

P. O. Box 68 (Pietari Kalmin katu 5)
00014 University of Helsinki

HELSINGIN YLIOPISTO — HELSINGFORS UNIVERSITET — UNIVERSITY OF HELSINKI

Tiedekunta — Fakultet — Faculty	Koulutusohjelma — Utbildningsprogram — Degree programme
Faculty of Science	Master's Programme in Data Science

Tekijä — Författare — Author

Rick Joosten

Työn nimi — Arbetets titel — Title

Discourse Act Classification in Asynchronous Online Forum Discussions

Työn laji — Arbetets art — Level	Aika — Datum — Month and year	Sivumäärä — Sidantal — Number of pages
Master's thesis	September 21, 2020	44

Tiivistelmä — Referat — Abstract

In the past two decades, an increasing amount of discussions are held via online platforms such as Facebook or Reddit. The most common form of disruption of these discussions are trolls. Traditional trolls try to digress the discussion into a nonconstructive argument. One strategy to achieve this is to give asymmetric responses, responses that don't follow the conventional patterns. In this thesis we propose a modern machine learning NLP method called ULMFiT to automatically detect the discourse acts of online forum posts in order to detect these conversational patterns. ULMFiT fine-tunes the language model before training its classifier in order to create a more accurate language representation of the domain language. This task of discourse act recognition is unique since it attempts to classify the pragmatic role of each post within a conversation compared to the functional role which is related to tasks such as question-answer retrieval, sentiment analysis, or sarcasm detection. Furthermore, most discourse act recognition research has been focused on synchronous conversations where all parties can directly interact with each other while this thesis looks at asynchronous online conversations. Trained on a dataset of Reddit discussions, the proposed model achieves a matthew's correlation coefficient of 0.605 and an F1-score of 0.69 to predict the discourse acts. Other experiments also show that this model is effective at question-answer classification as well as showing that language model fine-tuning has a positive effect on both classification performance along with the required size of the training data. These results could be beneficial for current trolling detection systems.

ACM Computing Classification System (CCS):
Computing methodologies → Artificial intelligence → Natural language processing
→ Discourse, dialogue and pragmatics

Avainsanat — Nyckelord — Keywords

NLP, Discourse Act Classification, Trolling, Transfer Learning

Säilytyspaikka — Förvaringsställe — Where deposited

Muita tietoja — Övriga uppgifter — Additional information

Contents

1. Introduction

An increasing amount of people are engaging in discussion on social media platforms. It is common for online trolls to disrupt these discussions. A troll inserts itself into the conversation and tries to digress conversations into a series of nonconstructive series of comments by posting incoherent comments regarding to the current discussion [23]. In this thesis we try to automatically detect how a discussion proceeds by applying modern natural language processing methods to automatically label each post in a discussion with its discourse act.

Common patterns of discourse employed by human trolls are the use of asymmetric responses [42]. A regular conversation consists of structural units called adjacency pairs which consists of two parts. The first part (e.g. a question) creates an expectation for the second part (an answer). Trolls often subvert this expectation by responding with an unexpected second pair part such as replying to a question with a question or ignoring the first pair part altogether. The example below shows a troll (Aaron Aaronson) starting with a statement to digress the conversation and purposely misunderstands the reply as a request of information while it is actually an accusation thus expecting either admission or denial of the accusative statement as the second pair part.

Identifying these asymmetric response patterns could be an effective way to detect trolls in online discussion spaces. Current trolling detection systems are machine learning models that primarily use thread level meta-information for classification such as whether posts are on-topic, the level of controversy of a post as well as the content of each post [49]. This behaviour has been observed in human trolls, it is expected that at least a subset of troll bots behave similarly. Troll bots designed to shape public opinion around topics covered by journalists exhibit similar behaviour as human trolls such as name calling and purposefully posting about controversial or taboo topics [27]. However, detecting troll bots is often more successful using user behaviour compared to post content [35, 27]. Discourse act asymmetry in online discussions seem to be a good indicator of human trolls. Automatically detecting this asymmetry and using them as an additional feature in new troll detection systems could improve the performance of these systems. However, this is a hard problem since it requires a robust system for

1	**Aaron Aaronson**:	There are two kinds of people in the world. People who live with cats and people whose houses don't reek of cat piss.
2	Lagado:	Such a stunning insight. Did you come up with that all by yourself?
3	**Aaron Aaronson**:	The first part of the aphorism is quite common. The second part is an observation that a lot of people whose houses don't reek of cat piss tend to experience. So the answer to your question is yes and no.

note: Reprinted with permission from "How Internet trolls use asymmetrical response strategies to steer conversation off the track", by H. Paakki, H. Vepsäläinen, and A. Salovaara, 2020 [42].

discourse act recognition (DAR).

Current research on discourse act recognition is mainly focused on transcriptions of real conversations between two or more participants or in online text conversations. The asynchronous nature of online forum discussions is a different environment to which current DAR models need to adapt. Besides the asynchronism of the conversation, context information given by intonation and non-lexical utterances like filler words ("hmmm") are lost.

In this thesis we try to create a model which overcomes these challenges by using modern LSTM based approach to classification combined with a transfer learning approach to create an accurate language model for the online conversation domain. Recent studies have shown that LSTMs are efficient in NLP tasks for classification of sequences [15, 26, 36]. By fine-tuning the language model with Reddit conversations from the coarse discourse dataset [53] using the ULMFiT method [26], the model learns a more accurate language representation for the online discussion domain.

Furthermore, this method could reduce the amount of training data needed to train such a model. This is important since manually labeling training data is a time-consuming and laborious task. Big modern models based on transformers such as BERT [12] require ten thousands of training examples to perform well. The smaller model proposed in this thesis provides a more practical solution to for when training data or computational power is limited.

First we will present a theoretical background of trolling, discourse act recognition and deep learning methods as well as a discussion on modern NLP methods. This is followed by a detailed description of the methodology how the ULMFiT methods works

in chapter 3.

Subsequently three experiments are discussed. The first experiment showed that the proposed model for DAR performs close to state of the art using a dataset of approximately 99 000 Reddit posts each manually labelled to be of one of nine discourse acts [53]. The second experiment shows that the proposed model also performs well in question-answer classification. This task is similar to the first experiment but instead of 9 classes the model classifies posts into three classes, question, answer, or other. The final experiment will show that the language model pre-training indeed improved the model's performance as well as reduces the number of training data needed.

2. Theoretical background

This section will give an overview of the main theoretical background which forms the foundation of this thesis. It starts with trolling and discourse act recognition before moving on to a overview of neural networks working from general theory towards more specific variants and techniques used in this thesis.

2.1 Trolling

Trolling is antisocial behaviour created by luring participants of the discussion into a fruitless discussion. Contrary to more modern politically motivated trolls, the traditional troll does not cause harm on a large scale, often a troll is motivated by their own amusement and has no further agenda [22]. A political troll wants to mobilize people to achieve a common goal set by the troll either overtly or covertly. This can be done by creating opportunities for their allies. However, often it is about destroying opportunities for their opposition [19]. Furthermore, political trolls often mobilize the people around them to join their cause and even employing tools like creating bots, while the traditional troll often acts alone.

Other research suggests that traditional trolls are not only people who set out to engage in trolling behaviour but everyone can exhibit trolling behaviour influenced by their mood and seeing other trolls in the discussion [8]. The anonymity of online forum based discussion and the ambiguities it creates allows the troll to digress the conversation. The main source of ambiguity anonymous discussion creates arises due to the uncertainty of the motivations behind each post as well as no information about the tone of voice [24].

Unlike online harassment and cyberbullying, trolling does not target a specific individual but the all participants in a discussion. Cyberbullying is intentional behavior to harm another person by repeatedly harassing the victim which is in a difficult position to defend themselves. The defencelessness often stems from a power imbalance between the harasser and the victim [47].

This thesis is primarily focused on the traditional troll using Hardaker's definition of trolling [23]. Hardaker classified the six main trolling strategies of this type

of trolling. These are: *Aggress trolling*, deliberately aggressive behaviour luring participants into retaliation. *Shock Trolling*, posting about taboos within the community. *Endanger trolling*, giving poor advice while pretending to be innocent. *Antipathy trolling*, creating a antagonistic environment in which it can be purposefully provocative. *(hypo)critical trolling*, being extremely critical of others. *digress trolling*, creating off topic discussions by lowly introducing tangential topics.

Using this definition of trolling, closer analysis of the trolling strategies has shown that human trolls deliberately use asymmetrical response patterns [42]. Symmetrical pairs would for example be: question–answer, request for action–acceptance/rejection, accusation–admission/denial, assertion–assertion. Not following these standard conversation patterns is improper behaviour in a normal conversation which trolls can exploit.

2.1.1 Trolling Detection

Trolling detection is the act of automatically detecting users who exhibit trolling behaviour in an online conversation. While this thesis does not concern a trolling detection system, it is intended to give insights for future systems.

Current trolling detecting systems usually work on one of three levels: post level, thread level, and community level. Post-based methods usually detect trolls using machine learning methods such as sentiment analysis to find malicious users [49]. Other approaches are related to finding posts with a low readability score [14]. Other methods based solely on metadata such as the length of a comment, number of aggressive words, key words, and the author perform not as well as ML methods. However, combined they provide better results than each one separate [21].

Thread-based approaches are methods that attempt to identify trolls on a thread level and not by single comments. It often involves techniques from post-bases methods but more information such as whether a post is on-topic compared to the rest of the discussion, or the degree of controversy of a post within the thread can be included in the detection [49].

Lastly, community-based methods take an even more high level approach. They apply techniques from social network analysis to users within a community. These often revolve around centrality and popularity measures. These can be used to create a reliability measure of each user within a community where trolls perform poorly [41].

2.2 Discourse Act Recognition

Discourse act recognition (DAR) is the problem of automatically detecting the structure of a conversation. Usually the goal is to label an utterance in a conversation with the intention of the user. The definition of what a discourse act (DA) is can vary depending on the goal of the model [32].

The classic definition of DA is the function of a sentence in the dialogue [3]. For example, generally a question has the function of requesting information. In this case the DA would be "request of information".

In the context of natural language understanding, discourse acts are often defined by the intent of utterance dependent on the application. For example, for a helpdesk chatbot DAs could be "show information", or "search address" giving the bot specific tasks dependent on the DA [29].

In conversation analysis (CA), a sociological approach of analysing casual conversation, DAs are defined similarly to the classic definition. However, it is not the function of the utterance, but the more general form of the utterance such as "question", "answer", or "positive reaction" that is used as a DA [32]. There have been attempts at standardizing the DA for CA [11, 7]. However, no standard has been adapted [15].

Because of the asynchronous nature of online textual conversations, tagging for natural language does not directly translate. Online conversation DA are often a combination of function and general form.

DAR requires disambiguation of utterances between multiple possible discourse acts. This is a non-trivial task, even for humans. Classic examples of ambiguous utterances are hypothetical questions and sarcasm. Both have the syntactical structure of one DA (question and statement) but actually serve a different function (statement/call for action and disagreement/humor respectively). In these cases the context in which the utterance was made plays a crucial role. Textual conversations present an even greater challenge since the speaker's intonation is completely lost.

Most modern DAR approaches use machine learning techniques. These generally consist of two main parts, a language model (LM) and a classification part. The LM is responsible for creating a numerical representation of written text such that it can be used by a computer. These models are trained by using a large corpus to predict the next token of a sequence, given all previous tokens. This creates a probability distribution over strings of texts. Based on this a model can place each word in a high dimensional vector space and use this vector representation as the input feature for classification. To create representations of sequences often include averaging the word vectors but various other techniques have been explored [45].

The classifier uses these representations as its input and is trained using a labelled dataset. These labels are the ground truth about the DA given by human annotators. After training it has learned to output the most likely label for each given input. This is discussed in detail in the next section and in section 3.3.

Most work on DAR has been focused on either a limited domain or a limited set of DAs such as question-answer pairs. These include, DAR on online debates on specific topics [13], classifying queries and solution on online help forums such as Tripadvisor or MOOC forum posts [2, 5].

In order to capture more of the context tree-based LSTM models which models group chat conversations as trees whose structure is encoded in an directed-acyclic-graph LSTM [54]. The first DAR model for online asynchronous textual conversations on a variety of topics with high level DA labels for each comment was proposed by Zang et al. [53]. They proposed a CRF model that used both the textual content as well as various structural and context features. It performed well with all features but struggled without them. This makes this model work well on discussions of a similar structure as their training data which was from Reddit. To resolve this issue Dutta et al. [15] proposed an attention based hierarchical LSTM to predict discourse acts without platform dependent structural features. This model performs well on a more general dataset but requires much labeled training data which is often not available. Transfer learning methods could help improve the model quality on smaller datasets and seem to perform well in more general text classification tasks [39].

2.3 Recurrent Neural Networks

Recurrent Neural Networks (RNN) are the main tool used in this thesis. RNNs are recurrent because they perform the same function for every input of data, while the output at time step t depends on that of it at $t-1$. This allows the model to pass information learned from one input to the next. In traditional feed forward neural networks each input is processed independent from each other while in RNNs the inputs are related in a sequence to each other. Figure 2.1 shows how the output of one iteration is recurrently passed to the next iteration. This allows for a more accurate representation for when inputs are independent from each other and follow some sequence in time or space [33].

This is useful for DAR since text data is a sequence. In a word level model the information gained by the previous words is passed in giving each input some context. Furthermore, RNNs allow the inputs to be of variable lengths by forwarding the last element of the sequence trough a softmax layer to make a prediction. Other models are built such that the input has to be of a fixed size which online forum posts are not.

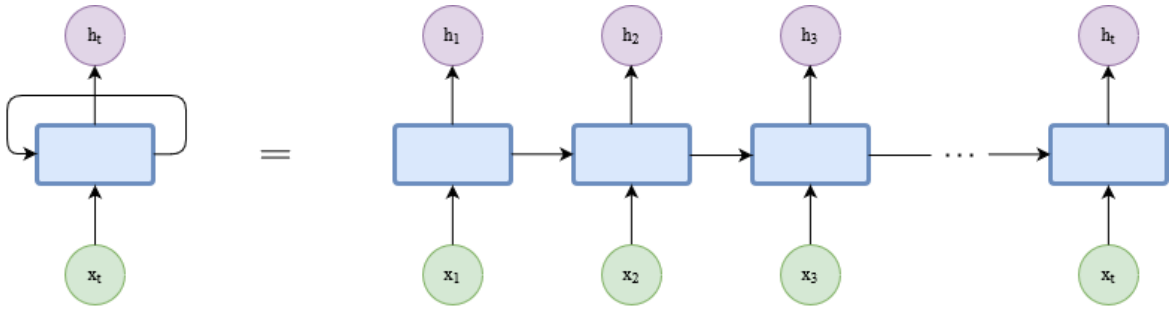


Figure 2.1: Recurrent neural network (left) and an unrolled RNN (right)

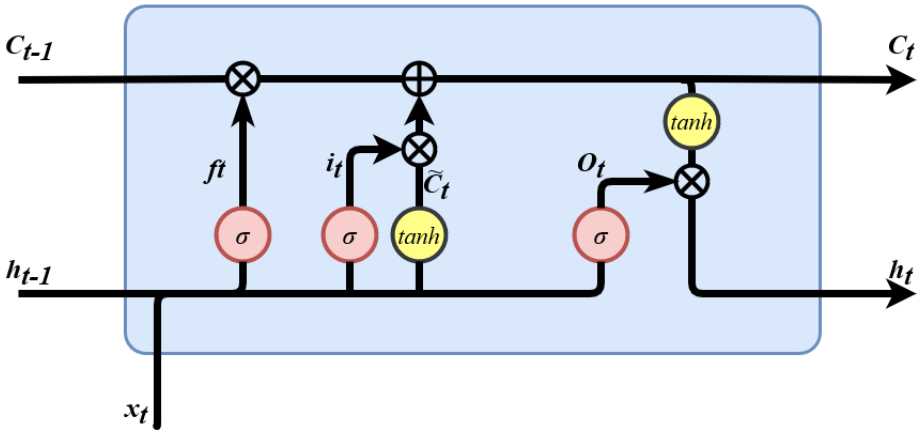


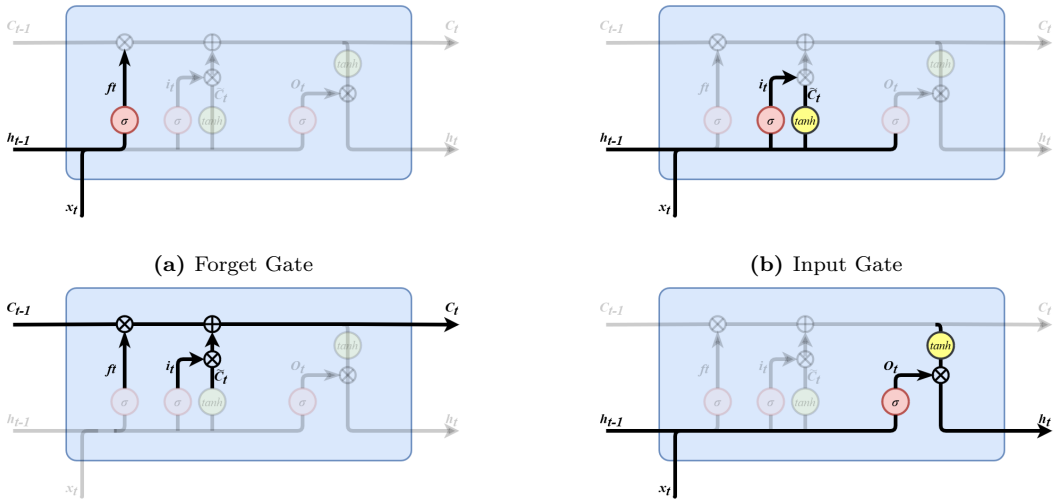
Figure 2.2: Cell state of an LSTM

Lastly, using deep learning language models allows for a more precise dense language representation compared to representations that are feature engineered by hand or vector representations based on word frequency and co-occurrence [30].

2.3.1 LSTM

While the basic RRN described in the previous section passes on information to the next state, it struggles representing long term dependencies. This is due to exploding or vanishing gradients. To solve this problem an extension of the basic RNN was invented named Long Short-Term Memory (LSTM) [25]. The main idea behind an LSTM is that it adds a cell state to a vanilla RNN which "remembers" information (Line labeled C in Fig. 2.2). At each time step this cell state is updated using three gates, a forget gate, an input gate, and an output gate.

At each time step t the model goes through the following steps (also see Fig. 2.3):



(a) Forget Gate **(b)** Input Gate

(c) Updating Cell State **(d)** Output gate

Figure 2.3: Inner workings of an LSTM: Forget Gate, Input Gate, and Output gate.

$$
\begin{aligned}
f_t &= \sigma(W_f x_t + U_f h_{t-1} + b_f) \\
i_t &= \sigma(W_i x_t + U_i h_{t-1} + b_i) \\
\tilde{C} &= tanh(W_C x_t + U_C h_{t-1} + b_C) \\
C_t &= f_t \circ C_{t-1} + i_t \circ \tilde{C}_t \\
o_t &= \sigma(W_o x_t + U_o h_{t-1} + b_o) \\
h_t &= o_t \circ tanh(C_t)
\end{aligned}
\tag{2.1}
$$

with W and U being weight matrices for the input and recurrent connections respectively, σ a sigmoid activation function, and b the bias term. First, the forget gate f_t selects which information from the cell state to "forget". It looks at the new information x_t and the output of the previous cell h_{t-1} and multiplies each element of the cell state C by a value between 0 and 1 created by the sigmoid activation function (Fig. 2.3a). Next a sigmoid layer called the input gate i_t decides what new information to update the cell state with and a new vector $\tilde{C}$ is created containing new possible values for the cell state (Fig. 2.3b). This is multiplied by i_t and used to update the cell state (Fig. 2.3c). Finally, an output is created by the output gate o_t. First a sigmoid layer which uses the new information x_t and the precious cell output h_{t-1} is created to filter the cell state. This filter is then applied to the cell state which is put through a tanh and creates the output h_t (Fig. 2.3d).

By going through these steps the network can use information from previous cells in order to inform the output of the current cell. However, in most cases, not only the previous elements of a sequence are useful but also the following elements of a sequence. To this end a bidirectional model can be utilized. A bidirectional LSTM

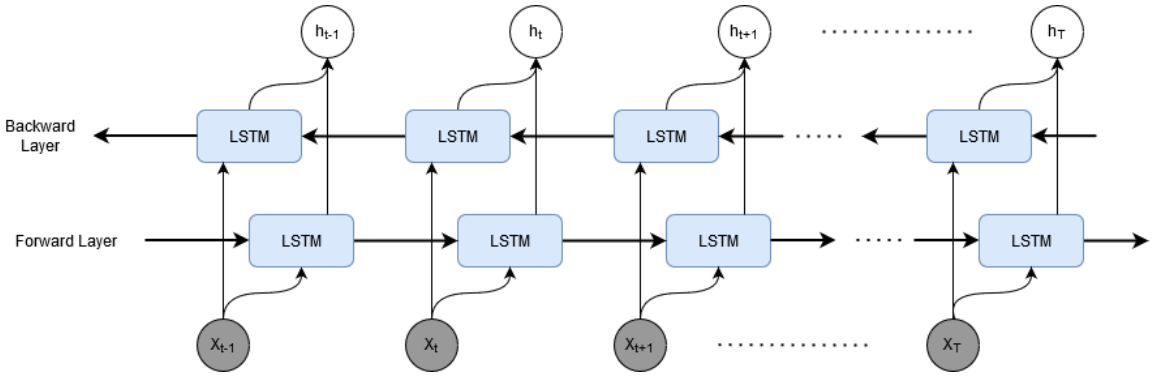


Figure 2.4: Bidirectional LSTM

uses two LSTMs in which one processes a sequence left to right and the other right to left after which the outputs are combined into a single output (see Fig. 2.4). In this way both the previous elements as well as the upcoming elements are considered.

2.3.2 AWD-LSTM

The Average-SGD Weight-Dropped LSTM (AWD-LSTM) is a further extension of the LSTM specifically designed with word level language modeling in mind [36]. AWD-LSTM applies several regularization methods and an addition to stochastic gradient descent without the need to fundamentally change how the LSTM works. Employing these strategies significantly increased the effectiveness of LSTM-based architectures for language modeling [4, 36]. These will be discussed in the rest of this section.

Weight-dropped LSTM

The cyclic nature of RNNs can easily lead to overfitting. Dropout has been a common technique to combat this. Dropout randomly changes the weight on some nodes to 0 meaning that they are not considered for that timestep. This dropout is often applied between timesteps to the hidden state vector h_{t-1} or on the cell state c_t [36]. However, this requires change to a standard LSTM structure meaning that highly optimized LSTM implementations, such as NVidiA's cuDNN LSTM cannot be used.

Instead, AWD-LSTM uses DropConnect, a generalization of dropout [51]. The basic principles of dropout are applied to the hidden-to-hidden weight matrices $[U_f, U_i, U_C, U_o]$ (see Equation 2.1) before the forward and backward pass. Because the same weights are used over multiple timesteps the dropped weights remain dropped for the entire forward and backward pass. This type of dropout does not change the workings of the LSTM such that standard implementations can be used.

Non-monotonically Triggered ASGD

Stochastic gradient descent (SGD) is the most common method for training neural networks. It iteratively takes steps to the optimum for a given loss function. This can be described by

$$\theta_t = \theta_{t-1} - \eta \cdot \nabla_\theta J(\theta) \tag{2.2}$$

with paramenters θ at time step t where η is the learning rate with $\nabla_\theta J(\theta)$ as gradient w.r.t. the objective function.

Averaged SGD (ASGD) takes the same steps as described by equation 2.2 but instead of returning the last time step, it returns an average given by

$$\frac{1}{K-T+1}\sum_{i=T}^{K}\theta_i. \tag{2.3}$$

K is the total number of iterations run and $T < K$ is an averaging trigger defined by the user when to start averaging.

In the AWD-LSTM, ASGD is expanded to counteract the need to fine tune the averaging trigger T. If the averaging is triggered too soon, the model might not be as effective. If the averaging is triggered too late the model might converge very slowly [36]. Instead of always averaging after a certain number of iterations it is possible to trigger it based on the performance of the model. The non-monotonically Triggered ASGD (NT-ASGD) introduces this principle by adding two additional hyperparameters, a logging interval L and an non-monotonic criterion n. Whenever the performance of the model does not increases for n iterations within the logging interval, averaging is triggered. This way the randomness of training a neural network does not influence when averaging is triggered, forgoing the need to fine tune the trigger T. Empirically it has been found that setting L to the number of iterations in an epoch and $n = 5$ works well for various models and datasets [36].

Further Regularization Techniques

Besides using a weight-dropped LSTM several other regularization techniques are implemented in AWD-LSTM to reduce the risk of overfitting. The rest of this section will give a brief overview of these techniques as presented by Merity et al. [36].

1. **Variable length backpropagation sequences**

   Backpropagation allows the network to update the weight matrices (W, U in Eq. 2.1) by minimizing the error between the output and the desired output using a loss function. This is the way the model learns. Training models for sequential data use backpropagation through time (BPTT) [43]. Variable length

backpropagation sequences prevents the problem of creating a partial backpropagation window by randomly selecting the sequence length for the forward and backward pass. This way the data is used more efficiently getting a full BPTT window while ensuring that the average sequence length remains around the base sequence length.

2. **Variational dropout**

   Instead of sampling a new dropout mask at every timestep, variational dropout samples a dropout mask once and repeatedly uses that dropout mask for all connections within a forward and backward pass.

3. **Embedding dropout**

   Dropout is applied on the word level embedding matrix. This is equivalent to removing all occurrences of a specific word within one specific forward and backward pass. The non-dropped-out embeddings have to be scaled by $\frac{1}{1-p_e}$ where p_e is the probability of embedding dropout.

4. **Weight tying**

   Weight tying shares the weights between the embedding and softmax layer in order to reduce the number of parameters in the model. It prevents the model having to learn a one-to-one mapping from input to output which improves performance of the LSTM [28].

5. **(Temporal) Activation Regularization**

   In order to control the norm of a model to reduce overfitting L_2-regularization is often used on the weights of the network. AWD-LSTMs apply 2 types of L_2-regularization, activation regularization (AR), and Temporal activation regularization(TAR).

   AR is L_2 regularization applied on the final hidden state output of the model. It adds a scaling coefficient α to the L_2 norm of the output state $h_t \circ m$, where m is the dropout mask. This penalises activations that are significantly larger than 0.

   TAR is L_2 regularization by adding the L_2 norm of the output state $h_t - h_{t+1}$ with a scaling factor β to the loss function. This regularization penalizes large changes in the hidden state.

2.4 Modern Language Models

In natural language processing (NLP) language models (LM) play a crucial role. Word embeddings are numerical vector representations of words often used as a feature for

various NLP tasks. There are different methods for learning these vector representations but most are learned in a self-supervised way. The goal of these representations is to have words which are semantically related represented close together in the vector space.

Using embeddings that were pre-trained on large unlabelled datasets such as word2vec [38] or GloVe [44] can be seen as a type of transfer learning. The goal of this is to transfer knowledge obtained from a more general task to a more specific task to improve its performance. The pre-trained word embeddings are used to initialise the first layer (embedding layer) of a model which is then trained to perform a specific task. While the first layer contains the most general information [52], pre-training a LM with the hidden layers can boost its performance [10, 26]. These types of pre-trained models capture more of the syntax and semantics and the usage of language in a general case. These models can then be fine-tuned to perform a specific task for a specific domain. The main benefit of using pre-trained LMs is that language data is easy to come by. However, labelled data for NLP tasks is often scarce which can be made up for by the aforementioned pre-training.

Recent models using these techniques are ELMo [46], BERT [12], ULMFiT [26], SiATL [10] and various variations on these. ELMo embeddings are obtained from LMs and then fine-tuned using additional contextual representations of the text. This approach requires a different architecture for different tasks instead of having a generic architecture applicable to different tasks [10].

BERT [12] pretrains LMs and fine tunes them on the specific task. To improve the language representations BERT employs next sentence prediction. Since BERT is a large transformer model which is fined-tuned with masked, bidirectional LMs it requires large amounts of computational resources as well as much training data.

ULMFiT is a transfer learning model which can be trained only using one GPU. It is trained using a three step process: LM pre-training, LM fine-tuning using domain specific language, and classifier fine-tuning where the model is trained to perform a specific task. It uses transfer learning techniques from computer vision where the model is pre-trained and fine-tunes the last (or several of the last) layers while keeping the earlier layers frozen. Chapter 3 will describe ULMFiT in more detail.

Lastly, SiATL (Single-step Auxiliary loss Transfer Learning) is a model closely related to ULMFiT. As the name suggests, it employs an auxiliary loss function in order to compress LM fine-tuning and classifier fine-tuning into a single step [10].

3. Methodology

This chapter covers the methodology used for classifying forum users posts as different discourse acts. First, a formal definition of the classification problem is given. Next, the preprocessing steps are described followed by a detailed description of the ULMFiT transfer learning model. Finally the evaluation metrics used to asses the quality of the qualifier are discussed.

3.1 Problem Statement

A post X is a sequence of tokens from some alphabet A of length n. Given this alphabet A, post X can be defined as:

Definition 1 $X = (x_1, \ldots, x_n)$, *where* $s_i \in A$ *for all* $i \in \{1, \ldots, n\}$.

For example: $X =$ (I, do, n't, agree, with, you, !) is a post of length 7 with an alphabet A consisting of English words, word parts and punctuation marks.

Each post X has a discourse act label from the set C with a fixed number k unique labels $\{c_1, \ldots, c_k\}$. The problem is: given a post X, predict to which of the classes from C X belongs such that post-label pairs $\langle X, c_i \rangle$ can be formed.

3.2 Preprocessing

In order to use the data in our model it needs to be preprocessed. Before tokenization, the text was cleaned by using some regex expressions. These included, removing HTML characters, adding spaces around '/' and '#' characters, removing excess white space as well as making everything lowercase. In order to not lose information about the use of capital letters the 'xxmaj' and 'xxup' tokens are added before a capitalised words and before words in ALL CAPS respectively.

Using spaCy each text was tokenized and numericalized with a dictionary created from a wikitext-103 pretrained model [37]. Besides regular tokenization, punctuation

marks, line break characters are retained as well as negative words like don't and can't are split into parts 'do' and n't to retain information about the negation.

After tokenization a beginning of sequence (BOS) token, and an end of sequence (EOS) token are added to the start and end of a post. This is done to indicate that the sequence has ended and the next input token for the model does not relate to the previous tokens.

Lastly, in order to reduce the number of tokens repeated characters and word are replaced by a repeat token 'xxrep' and 'xxwrep' respectively and the number of times it is repeated. E.g. "lol, lol, lol, lol" is replaced with xx "xxwrep, 4, lol".

3.3 ULMFiT Model

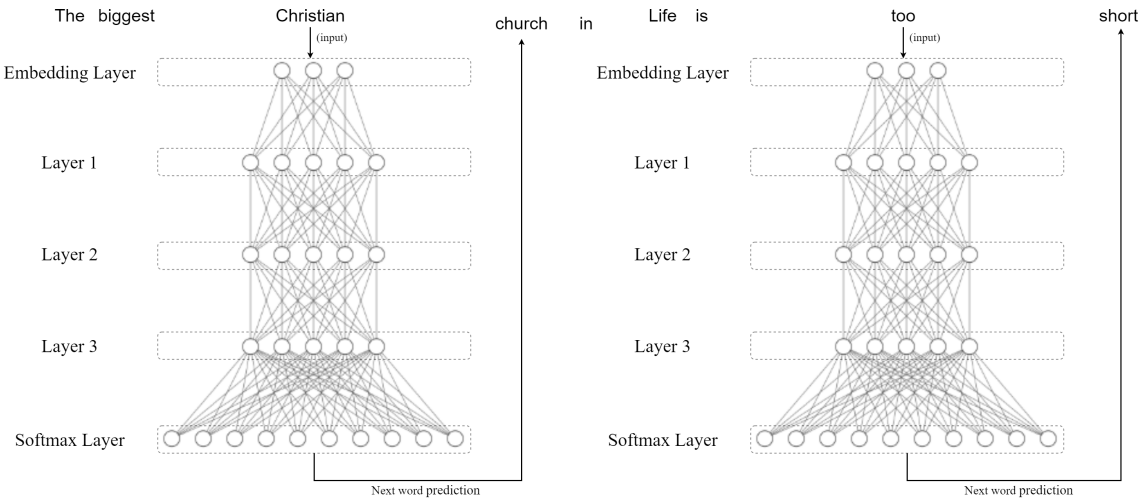
Universal Language Model Fine-tuning (ULMFiT) is a transfer learning model designed for NLP tasks [26]. ULMFiT performs well in text classification tasks along with requiring only a fraction of training examples compared to that needed for training transformer models [39, 26, 10]. It utilizes a 3-layer bidirectional AWD-LSTM structure combined with a three phase training strategy: LM pretraining, LM fine-tuning, and classifier fine-tuning (see Fig. 3.1). These stages as well as the transfer learning and learning rate optimization will be discussed in the following sections.

3.3.1 Language Model Pre-Training

The goal of general language model pre-training is to create a model that captures the general properties of language. Training a complete language model is very expensive therefore a pretrained LM on wikitext-103 [37] was used for this thesis. This model was trained using a continuous bag of word prediction, meaning it predicts the next token based on the context tokens. From this model only the embedding layer (or encoder) will be saved and used for the next phase. The part that is responsible for predicting the next word (decoder) is only used to train the embedding layer.

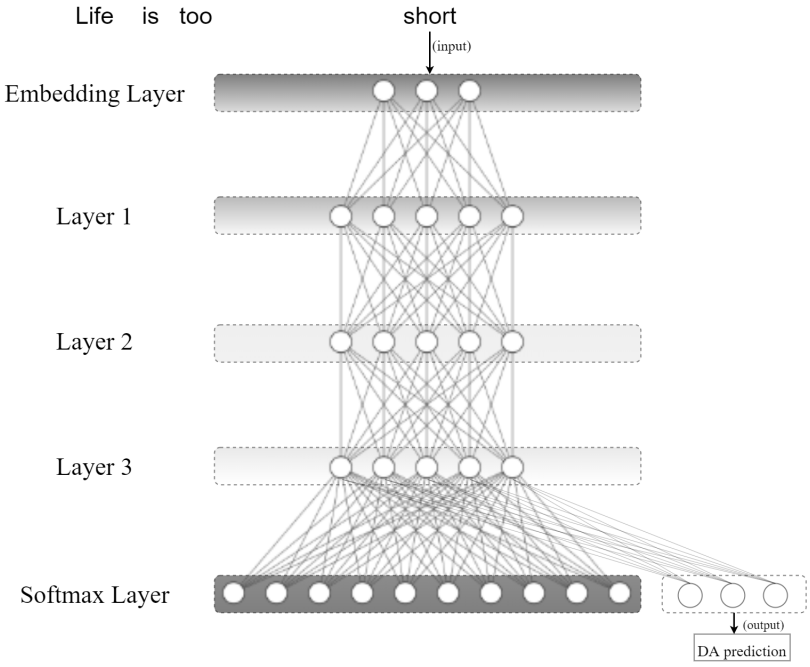
3.3.2 Language Model Fine-Tuning

To ensure good results on the target domain the language model trained in phase one is fine-tuned on target domain specific data. This way the LM can adapt to the specific properties of the language used in the target domain. The training strategy is the same as in phase 1 until the model is adapted to the target domain. While tokens that appeared in the general LM are kept the same, new tokens that did not appear during language model pre-training are initialized as the mean embedding of all the pre-trained embeddings.



(a) LM pre-training

(b) LM fine-tuning



(c) Classifier fine-tuning

Figure 3.1: Three stages of ULMFit: (a) Training LM on general-domain corpus (b) Fine-tuning LM on domain corpus test (c) Classifier fine-tuning using gradual unfreezing (darker parts are kept frozen for longer).

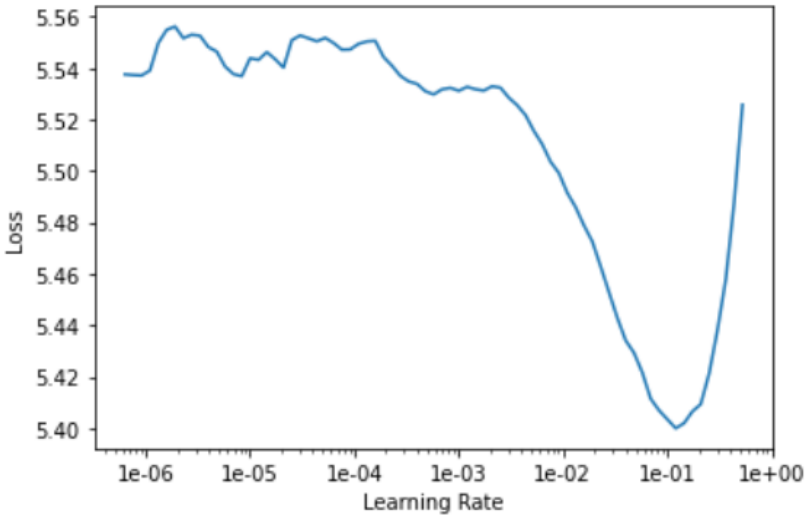


Figure 3.2: Validation loss per learning rate plot for learning rate selection

In addition to the learning process described for phase 1, ULMFiT uses discriminative fine-tuning and slanted triangular learning rates to fine-tune the LM.

Discriminative fine-tuning

In deep learning models different layers of the model capture different aspects of the data. Initial layers capture more general properties of the data while later layers are more attuned to the specific data the model is trained on [52]. Because of this, in order to fine-tune the model optimally, each layer should be fine-tuned to different extents.

Regular stochastic gradient descent is given by

$$\theta_t = \theta_{t-1} - \eta \cdot \nabla_\theta J(\theta) \tag{3.1}$$

with paramenters θ at time step t where η is the learning rate with $\nabla_\theta J(\theta)$ as gradient w.r.t. the objective function. In contrast, discriminative fine-tuning splits the parameters θ into $\{\theta^1, \ldots, \theta^L\}$ where L is the number of layers of the model. Equivalently, learning rate η is split up to obtain $\{\eta^1, \ldots, \eta^L\}$ such that η^l is the learning rate of the l-th layer. By splitting the parameters and learning rates the SGD update is the following:

$$\theta_t^l = \theta_{t-1}^l - \eta^l \cdot \nabla_{\theta^l} J(\theta). \tag{3.2}$$

The learning rates for each of the layers is set by choosing a learning rate for the last layer η^L and using $\eta^{l-1} = \eta^l / 2.6$. A starting learning rate of $1e-2$ was selected by validating the model while increasing the learning rate for several iterations. by plotting the loss w.r.t. the learning rate the learning rate where the loss decreases steeply was selected (see Fig 3.2).

Slanted triangular learning rates

When adapting the LM to the target domain for the target task, ideally the parameters would quickly converge to a suitable region of the parameter space. Instead of using the same learning rate (LR) for all iterations of training or an annealed learning rate, a slanted triangular learning rate [26] is used to achieve this. This is an adaptation of triangular learning rates where the LR first linearly increases and then linearly decays where the initial increase is quicker than the decay [48]. For this thesis the LR increases for the first 10% of training iterations.

3.3.3 Target Task Classifier Fine-Tuning

The third and final step is training a classifier using the fine-tuned LM. Two fully connected layers with ReLU and a softmax layer are added to the fine-tuned LM whose parameters are learned during this stage (see Fig. 3.1 c). After which it outputs a probability distribution over all classes.

Using only the final state of the LSTM for prediction could result in a loss of accuracy because ULMFiT is a word level recurrent neural network used for the task of text classification. The classification signal might only be a few words in any part of the text, this crucial information might be forgotten by the final LSTM state. To alleviate this problem concat pooling is used. The final state is concatenated with the max- and mean-pooled representation of the hidden states over as many time steps that fits in the training GPU's memory.

Fine-tuning the model for classification is the most crucial part of the process. Extremely aggressive fine-tuning can lead of catastrophic forgetting [31]. As a result, carefully learned parameters learned during the LM fine-tuning might be forgotten eliminating all their benefits. Conversely, overly cautious fine-tuning will result in slow convergence and overfitting. This is partially overcome by employing slanted triangular learning rates and discriminative fine-tuning. In addition to these ULMFiT adopts gradual unfreezing.

Instead of fine-tuning all layers at the same time, which could cause catastrophic forgetting, gradual unfreezing first fine-tunes the last layer for one epoch keeping the other layers frozen. Subsequently the procedure is repeated with the final two layers unfrozen and repeated again until all layers are fine-tuned. Early stopping was employed when the validation loss did not lower during training to avoid overfitting.

Using gradual unfreezing in conjunction with slanted triangular learning rates and discriminative fine-tuning has been shown to improve the error rates for text classification tasks [26] compared to other methods like chain-thaw where each layer is fine-tuned separately [17].

3.4 Evaluation Metrics

To evaluate the performance of the classifier the following evaluation metrics will be used: precision, recall, F1 score and Matthews Correlation Coefficient.

Precision

Precision is a metric given by the proportion of true positive (TP) predictions to all positive predictions. This gives a measure of confidence of the predictions.

$$Precision = \frac{TP}{TP + FP} \tag{3.3}$$

Recall

Recall is a measure given by the proportion of correct classification (TP) to the total amount of cases of the correct class. This metric can be seen as a measure of how complete the predictions are.

$$Recall = \frac{TP}{TP + FN} \tag{3.4}$$

F1 score

F1 score uses the harmonic mean of precision and recall in order to combine the two.

$$F1score = 2 \cdot \frac{precision \cdot recall}{precision + recall} \tag{3.5}$$

In case of mutliclass classification the F1 score for each class is calculated as if it was a binary classification problem and then either averaged (macro F1 score) or averaged weighted by the frequency of each class (micro F1).

Matthews Correlation Coefficient

Since F1 score does not take true negatives into account it can give misleading results when classes are imbalanced [6, 9]. Matthews correlation coefficient (MCC) is the correlation coefficient between the observed and predicted binary classifications given by:

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}. \tag{3.6}$$

In a binary classification case this returns a correlation value between -1 and +1 where a positive correlation indicates an agreement between the predictions and observations.

MCC has been generalized for a multiclass case which is also known as the R_k statistic [20]. This is in the discrete case given by:

$$R_k = \frac{\sum_{klm} C_{kk} C_{lm} - C_{kl} C_{mk}}{\sqrt{\sum_k (\sum_l C_{kl})(\sum_{l'k'|k' \neq k} \sum_{t'} C_{k'l'})} \sqrt{\sum_k (\sum_l C_{lk})(\sum_{l'k'|k' \neq k} C_{l'k'})}} \tag{3.7}$$

where C is a confusion matrix of size $K \times K$.

4. Experiments

In order to assess the effectiveness of the ULMFiT model for DAR three experiments are conducted. The first, main experiment tries to measure the effectiveness of three trained models at DAR with nine discourse acts. One model is a content only model while the other two also include structure features. This will measure the overall effectiveness of these kinds of models as well as evaluate the performance boost gained by including simple context features.

While DAR with all nine discourse acts is the main objective of this thesis, question-answer classification is an important task for information retrieval. The second experiment attempts to find out if a model with all nine classes is more effective at QA-classification compared to one that only classifies 3 classes using the models from the first experiment.

The last experiment tries to determine whether the pre-training approach of ULMFiT is useful. There are various machine learning approaches to DAR, most of them do not include a LM pre-training step. The effectiveness of this distinctive feature ULMFiT for DAR has to be measured. Furthermore, the performance of ULMFiT on smaller training datasets will be measured. An effective model for small datasets could be crucial for classification tasks in domains with limited labelled data.

In this chapter we will first describe and discuss the data used before moving onto the experimental setup, and the features used. Finally, the results of the three experiments are discussed and a brief summary is given.

4.1 Coarse-Discourse Dataset

The Coarse-Discourse dataset was released in 2017 by Zhang et al. [53]. It includes a randomly sampled selection of Reddit threads from its inception to May 2016. Threads from non-English, trading, and pornographic (NSFW) subreddits were removed. Furthermore, if a thread had fewer than two replies or contained deleted comments it was also excluded from the final dataset [53]. While these threads might have contained an unsuccessful trolling attempt, it is hard to verify without context whether it was a trolling attempt or not.

Discourse Act	Total %	Total Count
Answer	40.6%	40045
Elaboration	18.8%	18567
Question	16.3%	16047
Appreciation	8.6%	8439
Agreement	5.0%	4895
Disagreement	3.4%	3307
Humor	2.3%	2307
Negative Reaction	1.8%	1809
Announcement	1.4%	1367

Table 4.1: Percentage and count of posts per discourse act in the final dataset.

After the data was collected and filtered, they used 25 annotators to annotate each post with one of the following discourse acts: Question, Answer, Announcement, Agreement, Appreciation, Disagreement, Negative reaction, Elaboration, or Humor. Annotators were also allowed to assign a post to an Other category if it did not fit any of the nine discourse acts. Full definitions of the discourse acts are discussed in section 4.1.1.

The published data only contains a small number of metadata fields. Using these it is possible to retrieve the full text body of each post using the Reddit API. However, between the date of publishing and now some posts have been removed resulting in 14918 missing entries. Furthermore, 1308 posts were retrieved but contained the deleted marker ([deleted]) as well as 1173 comments without a body. These posts were also removed from the dataset. Upon closer inspection, the data also contained 291 entries from obvious bot accounts which are used on Reddit for quality of life improvements such as a bot that automatically links to a wikipedia page about the current discussion topic. These posts were also removed.

Finally the dataset contained 98667 posts with the following fields: a unique ID for each post, author name, full text of post, depth of post in thread, binary feature for whether a feature is the first post in a thread, discourse act annotation.

4.1.1 Discourse Act Definitions

As discussed above, each posts has been assigned one out of 9 discourse acts by human annotators. The following list gives definitions for each DA as defined by Zhang et al. [53].

Question: A question or request seeking a response with information, help, or feed-

back. Not every comment with a question structure is considered a question, such as rhetorical questions.

Answer: A comment in response to a question which gives the information the question asked for. An answer is always paired with a question.

Announcement: A comment presenting new information to the forum's community. For example, this can be in the form of news, a story, or a review. It is generally not linked to another comment.

Agreement: A post expressing agreement with a previous comment. This can be in the form of acknowledgement was well as providing additional evidence to the point made.

Appreciation: Comments expressing agreement, excitement, or praise in relation to another comment. In contrast to an agreement or elaboration, appreciation does not provide more information.

Disagreement: A comment correcting, criticizing, objecting to information given in a previous comment. The disagreement discourse act can also be used to provide evidence to support why the person disagrees.

Negative Reaction: comments expressing a negative reaction to a previous comment without discussing the merits of the previous comment or trying to disagree.

Elaboration: A comment which simply adds more information to a previous comment, this can be clarification of a question or simply adding more information to an answer.

Humor: A comment which is intended as a joke. This can be sarcasm, a pun, or a silly comment without adding more information. Sarcasm which tries to make a point or is mean spirited should be considered disagreement or negative reaction.

4.2 Experimental Setup

All experiments were set up using the ULMFiT setup with a three layer bidirectional AWD-LSTM as described in chapter 3. If not stated otherwise, the following set of hyperparameters were used for each experiment. The AWD-LSTM language model is used with an embedding size of 400 and 1150 nodes per hidden layer and hyperparameters for the ASGD as recommended by Merity et al. [36].

Epoch	Trained Layers
1	3
2	3,2
3	3,2,1
4	3,2,1
5	3,2,1, embedding
6	3,2,1, embedding
7	3,2,1, embedding

Table 4.2: Trained layers during each epoch of training using gradual unfreezing.

Default dropout rates are applied in line with the experimental setup of Howard and Ruder [26]: 0.4 to layers, 0.3 to RNN layers, 0.4 to input embedding layers, 0.05 to embedding layers, and weight dropout of 0.5 to the RNN hidden-to-hidden matrix. All dropout was then multiplied by 0.6 which empirically gave the best results for both language model fine-tuning and classifier fine-tuning.

For classifier fine-tuning two fully connected layers were added to the fine-tuned language model with 50 nodes in the hidden layer. Adam was used as optimizer with categorical cross entropy as loss function. Using the method described in section 3.3.2 the base learning rate was set to 0.02 for both fine-tuning the LM and the classifier.

For the first two experiments the data was randomly split into a 80-20 train-test split. Subsequently the test data was randomly split into a 70-30 test-development set. These splits were made at the comment level meaning that it is possible that comments from the same discussions could end up in any of the three sets. The LM was fine-tuned once using all datapoints, saved, and subsequently used for all three experiments.

The experimental setup for full DAR and Q&A classification are the same. The only difference is that for Q&A classification the labels were changed into "other" if the original DA label was not "question" or "answer". First, the fine-tuned LM was loaded after which the model was trained with gradual unfreezing for 7 epochs. Table 4.2 shows which layers we trained during each epoch, the layers are labels as in figure 3.1. The stopping criterion of 7 epochs was found experimentally and selected when the validation loss did not decrease for 2 epochs. Longer training resulted in an increase in both training and validation loss. All models were trained via google colab on a NVIDIA Tesla K80 GPU with a batch size of 32.

The third experiment applied the same training with gradual unfreezing techniques as the first two experiments except the number of epochs was selected to be the number of epochs until the validation loss did not decrease for 2 epochs in a row instead of the fixed 7 epochs. Figure 4.1 shows how the data was utilized. First the

Figure 4.1: Train-test splits of the data for experiment 3. A fixed test set was used with different percentages of the training set.

data was randomly split into a 80-20 train-test split. For each subset of this dataset a random sample from the first training set was taken to make up the other percentages of training data. The test set was kept constant. The other data was not used. For each percentage of training data two models were trained: one with loading the fine-tuned LM, the other with loading the wikitext-103 language model.

4.3 Features

While the main focus is on content based discourse act classification, it has been shown that including context based features generally improves classification [15, 53, 34]. Out of several context features, including features related to the structure and position of a Reddit comment leads to the biggest increase in correct classification [53]. Because of this, these types of features will be included in the experiments.

Content The content consists of the preprocessed text of the comments. This includes words, punctuation, control characters and special tokens added during preprocessing (see section 3.2).

Structure Structure features are features related to the structure and position of the comment. These are: post depth and normalized post depth according to Reddit's tree-like comment structure, number of sentences, words, and characters in the body of the comment.

Model	Precision	Recall	Micro F1	Macro F1	MCC
All answers	0.16	0.41	0.23	0.06	*
Zang et al. [53] without context	0.54	0.56	0.51	†	†
Content only	0.62	0.65	0.62	0.45	0.521
Content + Depth	0.67	0.69	0.68	0.48	0.590
Content + Context	0.69	0.70	0.69	0.51	0.605

Table 4.3: Results for different models predicting discourse acts.

4.4 Results

The main results from the experiments are reported in Table 4.3. Predictions are made at the comment level with either only the content features, all context features, or only the content features and the depth feature. Two other metrics are used as a baseline. The first baseline is labeling all posts as the majority class "answer". The second is the results from Zang et al. [53]. They presented several experiments similar to these with similar context features as well. However, they concluded that the usefulness of the context features greatly depend on the forum. For this reason only the content-only results are used as the second baseline.

The ULMFiT model outperforms both baselines in each case. Figure 4.2 shows a breakdown of the matthews correlation coefficient per discourse act for the content and context model. Furthermore, the prevalence of each class is shown as well. It shows that the model performs decently for the agreement, appreciation and question discourse acts. As well as that classification of the "announcement", "elaboration" and "answer" discourse act are most benefited by the inclusion of the context features. This is not too surprising since the first post of a thread is generally an announcement or a question which is ordinarily followed by an answer.

Discourse acts such as "humor", "disagreement", and "negative reaction" have a relatively low MCC meaning the predictions and actual label are barely correlated. These classes however also have the lowest inter-annotator reliability score between the annotators of the original dataset [53] meaning that they were hard for humans to classify as well.

The confusion matrices in figure 4.3 show the normalized number of predictions for each class. The most common error both the content-only and content+context model make is misclassifying discourse acts as an answer. The inclusion of context features mainly improve the classification of "announcement", and "elaboration". This

*Cannot be calculated due to division by zero.

†Not reported

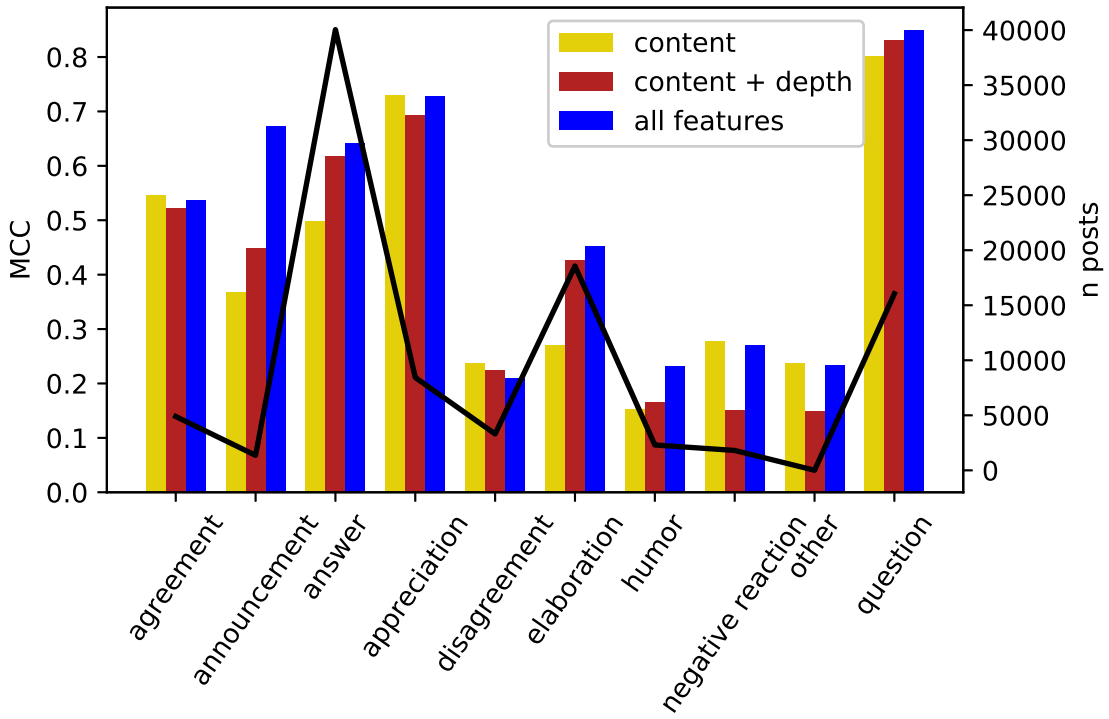


Figure 4.2: Matthews Correlation Coefficient of the three models broken down by discourse act along with prevalence in the data.

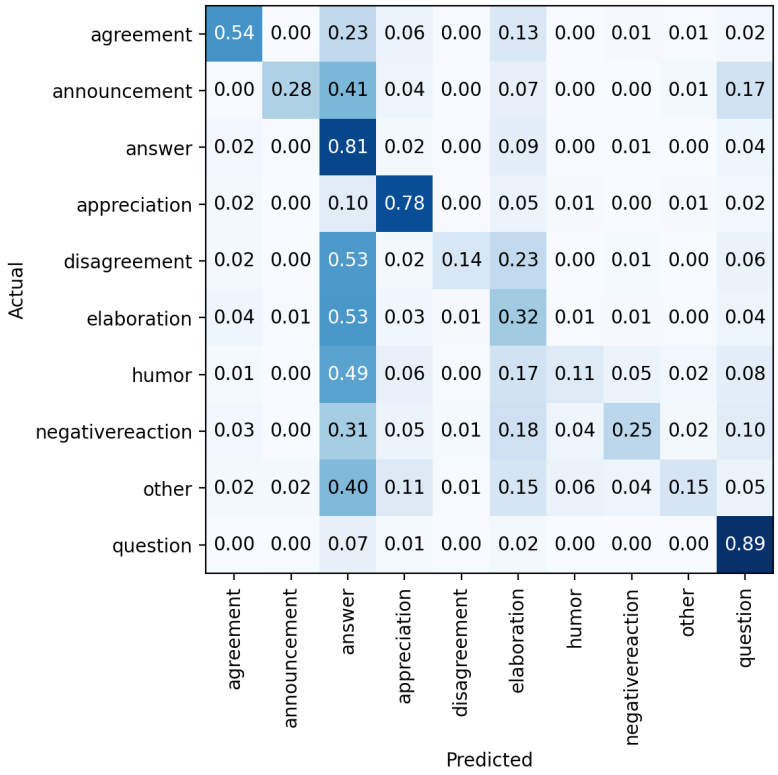
reduces the number of false positives of the "answer" class resulting in an increase in the MCC of "answer" as discussed above. However, the recall of predicting the "answer" discourse act stayed the same.

4.4.1 Q&A Classification

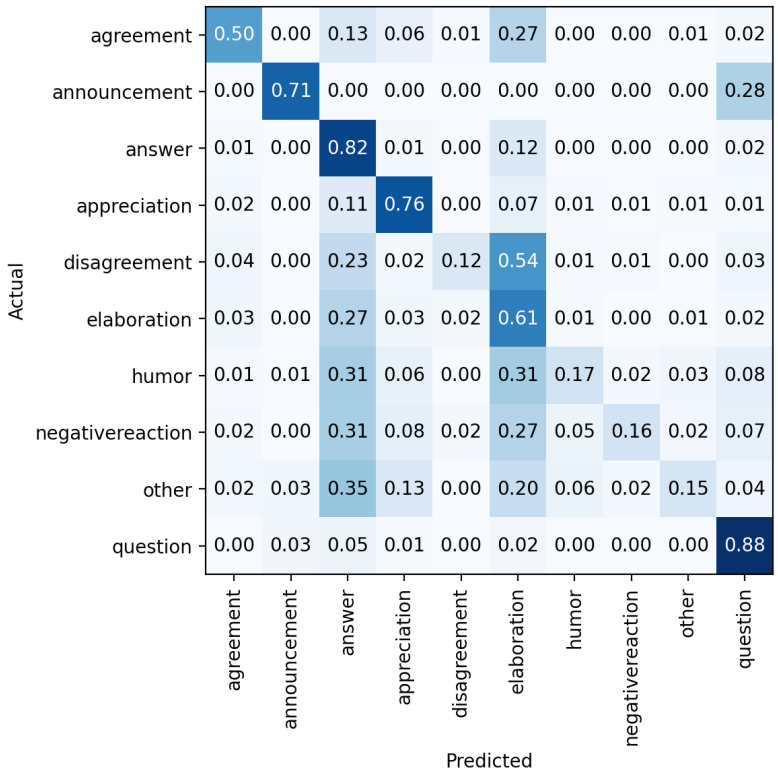
While the main focus is on classifying all discourse acts, the most common adjacency pair in Reddit discussions is question-answer [53]. Furthermore, Question-answer (Q&A) classification plays an important role in information retrieval. Most previous research has focused on classifying these two classes. However, the addition of other classes might worsen Q&A classification due to an increase in confusion between Q&A and the other labels.

To investigate this, Q&A classification was done using the current models with all 9 classes as well as Q&A prediction with 3 classes: "Question", "Answer", and "Other", where the non-Q&A classes were all considered "Other". Table 4.4 shows that the model preforms slightly better when there are only 3 classes as opposed to doing Q&A classification with the full 9 classes. This indicates that the additional classes indeed introduce more confusion between Q&A labels and the other labels.

Figure 4.4 shows the confusion matrix of the content + context model for Q&A



(a) Confusion Matrix Content only model



(b) Confusion Matrix Content + Context Model

Figure 4.3: Normalized Confusion Matrices Discourse Act Classification of two models

Model	Precision	Recall	Micro F1	Macro F1	MCC
Content only – 3 classes	0.73	0.73	0.73	0.76	0.573
Content + Depth – 3 classes	0.79	0.79	0.79	0.81	0.671
Content + Context – 3 classes	0.81	0.81	0.81	0.82	0.689
Content only – 9 classes	0.73	0.71	0.71	0.74	0.560
Content + Depth – 9 classes	0.79	0.79	0.79	0.80	0.660
Content + Context – 9 classes	0.80	0.80	0.80	0.81	0.679

Table 4.4: QA classification

classification. While the 3-class Q&A classification performs better overall, comparing this confusion matrix with the confusion matrix with all classes 4.3b it is clear that the 3-class prediction performs worse in terms of recall (specifically for the "Answer" DA) but shows improvement in precision.

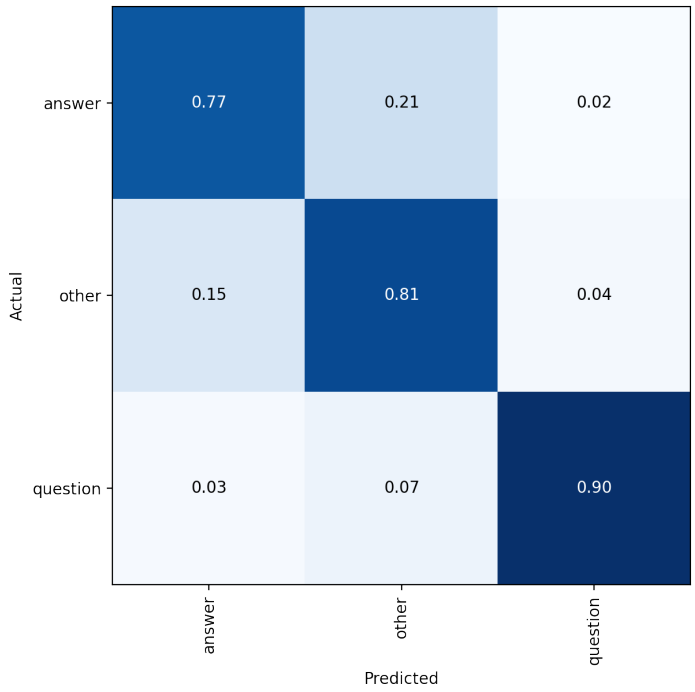


Figure 4.4: Confusion matrix of Question-Answer classification using Content + Context model.

4.4.2 Effect of LM fine-tuning

As discussed in chapter 2, applying transfer learning methods to DAR should reduce the amount of labelled training data needed to create a proficient model. The main idea behind ULMFiT is the three-stage process with the LM fine-tuning as the most novel aspect of it. To see what the effect of LM fine-tuning is on the number labelled

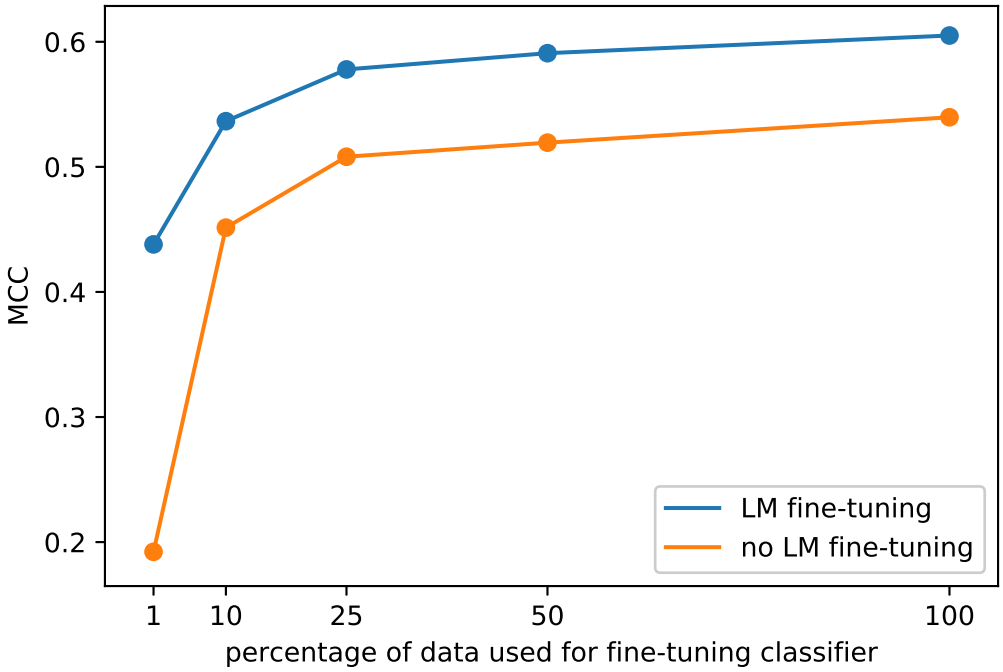


Figure 4.5: Matthews correlation coefficient for models trained with different amounts of training data ranging from 1% to 100% with and without language model fine-tuning.

data points needed is, several models were trained with different sizes of training data with and without LM fine-tuning. Figure 4.5 shows the results of how LM fine-tuning affects the performance of a model for various amounts of training data ranging from 1% (790 samples) to 100% (79094 samples).

LM fine-tuning has an overall beneficial effect on the performance of the model, each model with LM fine-tuning outperformed the models without LM fine-tuning. Furthermore, the drop-off in performance in the models with lower amounts of training data is less steep for the model with LM fine-tuning than the model without. This indicates that during LM fine-tuning indeed the model learns valuable language information resulting in better performance.

4.5 Summary of Results

- ULMFiT produces close to state of the art results on DAR with a content only model.
- Including context features improved the performance of the model.
- "humor", "disagreement", and "regative reaction" discourse acts were the hardest

to classify correctly.

- In Q&A classification, models classifying posts into 3 classes outperformed models classifying posts into 9 classes in terms of MCC.
- 3-class models had worse recall than 9-class models for Q&A classification.
- LM fine-tuning is overall beneficial for DAR performance.
- LM fine-tuning boosts performance more in small datasets compared to bigger datasets.

5. Discussion

This thesis shows that transfer learning using the ULMFiT method is an effective approach for discourse act recognition in online asynchronous Reddit conversations. At the comment level posts can be labeled with its corresponding acts without the need of much contextual information and without the need of large amounts of labelled training data due to the effects of language model pre-training. In addition to full DAR, the model also performed well in the subtask of question-answer classification. This application can be a useful feature in detecting asynchronous response patterns of trolling behaviour.

5.1 Limitations and Future Work

While the models perform well on a comment level, there is no information about the context on the thread level. Including more information about the context could improve the model [53]. One such approach is modeling the tree structure of the conversations with a DAG-LSTM [54] where the state of the LSTM predicting the previous comment is used to initialize the state of the current comment. While including more context features could improve DAR, using these kinds of models complicate the usage in online detection. While this model is designed for offline detection, models like these working on a comment level could potentially detect trolling in process. Including features requiring the discussion to be "complete" would make this impossible.

Another approach for improving DAR is improving the language representation. The current model models language at the word level. In order to capture pragmatic relationships between words there needs to be a representation of language at the comment level with word relevance attention depending on the context [15]. Combining this with transfer learning methods could produce state-of-the-art results for this task on small datasets.

The current model is only trained on an English language dataset. Since large parts of Reddit and other social media are non-English it is important that future models are multilingual. It has been shown that ULMFiT is a suitable method for modeling other languages [40, 50] as well as in multilingual models [16].

Furthermore, the data only consist of Reddit conversations. In order to generalize to asynchronous online conversation in general, other online conversation sources such as comment sections from news websites or Youtube need to be included. Currently, most DAR datasets consist of transcribed dialogues between humans, in which the synchronicity of the conversation plays an important role, or datasets from online chats between users. Multi-user asynchonous forum conversations are often not investigated from a DAR perspective.

Lastly, many current data science approaches are purely data driven. Insights from sociolinguistics and conversation analysis might inform different model structures or reveal relevant features to include in future models. However, in other domains it is shown that given enough data, a complex enough model can produce outstanding results. One such example is Meena, an open-domain chatbot being able to understand and produce human-like language trained with 867M messages [1].

5.2 Applications

The main intended application intended for this DAR model is to be used as a feature for trolling detection. Previous research suggests that trolls on online forums use asymmetric response patterns to derail a discussion [42]. Using DA classification to label posts with its DA, asymmetric patterns can be detected potentially increasing the effectiveness of trolling detection models. Furthermore, a robust trolling detection system could further create more insights into the structure of dialogue acts on a thread level in trolling behaviour.

Finally, many current chatbots and service bots rely on a robust DAR system in order to understand queries presented by a user. A well-performing model can increase the effectiveness of the language understanding of these bots. Transfer learning can especially be beneficial for bots operating on a specific domain of which not much training data is available.

5.3 Ethical Considerations

While social media posts are often publicly available, using it does require some ethical considerations. First of all, most social media users do not have an expectation that public social media data is used for research [18]. Since the data is randomly sampled from several Reddit forums (subreddits) it is a possible risk that a user shared sensitive information. In this case it is ethically responsible to obtain informed consent from the user in order to use the data, even though it is public. While other platforms such as Facebook and Twitter have clear guidelines for limitations for publishing and

anonymizing public data from their platform, Reddit does not. It is always a good practice to anonymize posts when presenting them if it is not at odds with the user agreement.

While it does not directly apply to this thesis, it is intended to use for a trolling detection model. In this case it is important to consider the benefits of publishing information about finding online trolls for the trolls themselves. Any new information considering how to combat online trolls can also benefit them by informing them what behaviour to avoid in order not to get detected or learn to use it to be more effective in their trolling. Furthermore, research into trolling behaviour exposes researchers to large numbers of negative comments which can affect them negatively.

Acknowledgements

This work was supported by the Academy of Finland, decision 314262.
Thank you to my supervisor Antti Ukkonen, examiner Indrė Žliobaitė, and all members of the FAKE research group.

Bibliography

[1] D. Adiwardana, M.-T. Luong, D. R. So, J. Hall, N. Fiedel, R. Thoppilan, Z. Yang, A. Kulshreshtha, G. Nemade, Y. Lu, and Q. V. Le. Towards a Human-like Open-Domain Chatbot. 1 2020.

[2] J. Arguello and K. Shaffer. Predicting speech acts in MOOC forum posts. In *Proceedings of the 9th International Conference on Web and Social Media, ICWSM 2015*, pages 2–11, 2015.

[3] J. L. Austin. How to Do Things with Words: Second Edition. page 192, 1975.

[4] S. Bai, J. Z. Kolter, and V. Koltun. An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling. *CoRR*, abs/1803.0, 2018.

[5] S. Bhatia, P. Biyani, and P. Mitra. Identifying the role of individual user messages in an online discussion and its use in thread retrieval. *Journal of the Association for Information Science and Technology*, 67(2):276–288, 2 2016.

[6] S. Boughorbel, F. Jarray, and M. El-Anbari. Optimal classifier for imbalanced data using Matthews Correlation Coefficient metric. *PLoS ONE*, 12(6):e0177678, 6 2017.

[7] H. Bunt. A methodology for designing semantic annotation languages exploring semantic-syntactic isomorphisms. In *Proceedings of the Second International Conference on Global Interoperability for Language Resources (ICGL 2010), Hong Kong*, pages 29–46, 2010.

[8] J. Cheng, M. Bernstein, C. Danescu-Niculescu-Mizil, and J. Leskovec. Anyone Can Become a Troll: Causes of Trolling Behavior in Online Discussions.

[9] D. Chicco and G. Jurman. The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. *BMC Genomics*, 21(1):1–13, 1 2020.

[10] A. Chronopoulou, C. Baziotis, and A. Potamianos. An Embarrassingly Simple Approach for Transfer Learning from Pretrained Language Models. pages 2089–2095, 2019.

[11] A. Clark and A. Popescu-Belis. Multi-level dialogue act tags. *Proc. SIGdial*, page 163â170, 2004.

[12] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference*, volume 1, pages 4171–4186. Association for Computational Linguistics (ACL), 10 2019.

[13] S. Ding, G. Cong, C. Y. Lin, and X. Zhu. Using conditional random fields to extract contexts and answers of questions from online forums. In *ACL-08: HLT - 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Conference*, pages 710–718, 2008.

[14] I. O. Dlala, D. Attiaoui, A. Martin, and B. B. Yaghlane. Trolls Identification within an Uncertain Framework. Technical report.

[15] S. Dutta, T. Chakraborty, and D. Das. How Did the Discussion Go: Discourse Act Classification in Social Media Conversations. pages 137–160. Springer, Cham, 2019.

[16] J. Eisenschlos, S. Ruder, P. Czapla, M. Kardas, S. Gugger, and J. Howard. MultiFit: Efficient multi-lingual language model fine-tuning. In *EMNLP-IJCNLP 2019 - 2019 Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing, Proceedings of the Conference*, pages 5702–5707. Association for Computational Linguistics, 9 2020.

[17] B. Felbo, A. Mislove, A. Søgaard, I. Rahwan, and S. Lehmann. Using millions of emoji occurrences to learn any-domain representations for detecting sentiment, emotion and sarcasm. Technical report.

[18] C. Fiesler and N. Proferes. âParticipantâ Perceptions of Twitter Research Ethics. *Social Media and Society*, 4(1), 1 2018.

[19] C. Flores-Saviaga, B. C. Keegan, and S. Savage. Mobilizing the Trump Train: Understanding Collective Action in a Political Trolling Community. Technical report.

[20] J. Gorodkin. Comparing two K-category assignments by a K-category correlation coefficient. *Computational Biology and Chemistry*, 28(5-6):367–374, 12 2004.

[21] J. Hallman and A. Lökk. Viability of Sentiment Analysis for Troll Detection on Twitter A Comparative Study Between the Naive Bayes and Maximum Entropy Algorithms. Technical report, 2016.

[22] C. Hardaker. Trolling in asynchronous computer-mediated communication: From user discussions to academic definitions, 7 2010.

[23] C. Hardaker. âUh. . . . not to be nitpicky„ „ ,but. . . the past tense of drag is dragged, not drug.â: An overview of trolling strategies. *Journal of Language Aggression and Conflict*, 1(1):58–86, 1 2013.

[24] S. Herring, K. Job-Sluder, R. Scheckler, and S. Barab. Searching for safety online: Managing "trolling" in a feminist forum. *Information Society*, 18(5):371–384, 10 2002.

[25] S. Hochreiter and J. Schmidhuber. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780, 11 1997.

[26] J. Howard and S. Ruder. Universal language model fine-tuning for text classification. In *ACL 2018 - 56th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference (Long Papers)*, volume 1, pages 328–339. Association for Computational Linguistics (ACL), 2018.

[27] J. Im, E. Chandrasekharan, J. Sargent, P. Lighthammer, T. Denby, A. Bhargava, L. Hemphill, D. Jurgens, and E. Gilbert. Still out there: Modeling and Identifying Russian Troll Accounts on Twitter. In *WebSci 2020 - Proceedings of the 12th ACM Conference on Web Science*, pages 1–10, New York, NY, USA, 7 2020. Association for Computing Machinery, Inc.

[28] H. Inan, K. Khosravi, and R. Socher. Tying Word Vectors and Word Classifiers: A Loss Framework for Language Modeling. 11 2016.

[29] M. Jeong and G. G. Lee. Jointly predicting dialog act and named entity for spoken language understanding. In *2006 IEEE ACL Spoken Language Technology Workshop, SLT 2006, Proceedings*, pages 66–69, 2006.

[30] D. Jurafsky and J. H. Martin. Vector Semantics and Embed-dings. In *Speech and language processing*, chapter 6. Prentice Hall, Pearson Education International, Upper Saddle River, NJ, 2nd edition, 2014.

[31] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska, D. Hassabis, C. Clopath, D. Kumaran, and R. Hadsell. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences of the United States of America*, 114(13):3521–3526, 3 2017.

[32] P. Král and C. Cerisara. Dialogue act recognition approaches. *Computing and Informatics*, 29(2):227–250, 2010.

[33] Z. C. Lipton, J. Berkowitz, and C. Elkan. A Critical Review of Recurrent Neural Networks for Sequence Learning. 2015.

[34] Y. Liu, K. Han, Z. Tan, and Y. Lei. Using Context Information for Dialog Act Classification in DNN Framework. Technical report.

[35] L. Luceri, S. Giordano, and E. Ferrara. Detecting Troll Behavior via Inverse Reinforcement Learning: A Case Study of Russian Trolls in the 2016 US Election. Technical report, 5 2020.

[36] S. Merity, N. S. Keskar, and R. Socher. Regularizing and optimizing LSTM language models. In *6th International Conference on Learning Representations, ICLR 2018 - Conference Track Proceedings*. International Conference on Learning Representations, ICLR, 8 2018.

[37] S. Merity, C. Xiong, J. Bradbury, and R. Socher. Pointer Sentinel Mixture Models. *5th International Conference on Learning Representations, ICLR 2017 - Conference Track Proceedings*, 9 2016.

[38] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean. Distributed representations ofwords and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, 2013.

[39] S. Minaee, N. Kalchbrenner, E. Cambria, N. Nikzad, M. Chenaghlu, and J. Gao. Deep Learning Based Text Classification: A Comprehensive Review. 4 2020.

[40] M. Ogrodniczuk and Å. Kobyliński. Universal Language Model Fine-Tuningfor Polish Hate Speech Detection. *Proceedings of the PolEval 2019 Workshop*, 2019.

[41] F. J. Ortega, J. A. Troyano, F. L. Cruz, C. G. Vallejo, and F. Enríquez. Propagation of trust and distrust for the detection of trolls in a social network. *Computer Networks*, 56(12):2884–2895, 8 2012.

[42] H. Paakki, H. Vepsäläinen, and A. Salovaara. How Internet trolls use asymmetrical response strategies to steer conversation off the track. *In preparation*, 2020.

[43] R. Pascanu, T. Mikolov, and Y. Bengio. On the difficulty of training recurrent neural networks. In *30th International Conference on Machine Learning, ICML 2013*, number PART 3, pages 2347–2355, 2013.

[44] J. Pennington, R. Socher, and C. D. Manning. GloVe: Global vectors for word representation. In *EMNLP 2014 - 2014 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*, pages 1532–1543, 2014.

[45] C. S. Perone, R. Silveira, and T. S. Paula. Evaluation of sentence embeddings in downstream and linguistic probing tasks. *CoRR*, abs/1806.0, 6 2018.

[46] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer. Deep contextualized word representations. In *NAACL HLT 2018 - 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference*, volume 1, pages 2227–2237. Association for Computational Linguistics (ACL), 2 2018.

[47] R. Slonje, P. K. Smith, and A. Frisén. The nature of cyberbullying, and strategies for prevention. *Computers in Human Behavior*, 29(1):26–32, 1 2013.

[48] L. N. Smith. Cyclical Learning Rates for Training Neural Networks. *Proceedings - 2017 IEEE Winter Conference on Applications of Computer Vision, WACV 2017*, pages 464–472, 6 2015.

[49] Tomaiuolo, Lombardo, Mordonini, Cagnoni, and Poggi. A Survey on Troll Detection. *Future Internet*, 12(2):31, 2 2020.

[50] B. van der Burgh and S. Verberne. The merits of Universal Language Model Fine-tuning for Small Datasets – a case with Dutch book reviews. *arXiv*, 2019.

[51] L. Wan, M. Zeiler, S. Zhang, Y. Lecun, and R. Fergus. Regularization of Neural Networks using DropConnect. Technical report, 2013.

[52] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson. How transferable are features in deep neural networks? Technical report, 2014.

[53] A. X. Zhang, B. Culbertson, and P. Paritosh. Characterizing online discussion using coarse discourse sequences. In *Proceedings of the 11th International Conference on Web and Social Media, ICWSM 2017*, pages 357–366, 2017.

[54] O. Ä°rsoy, R. Gosangi, H. Zhang, M.-H. Wei, P. Lund, D. Pappadopulo, B. Fahy, N. Nephytou, and C. Ortiz. Dialogue Act Classification in Group Chats with DAG-LSTMs. 2019.