Social Network Analysis and Mining (2020) 10:51
https://doi.org/10.1007/s13278-020-00665-4

ORIGINAL ARTICLE

Developing a Twitter bot that can join a discussion using state-of-the-art architectures

Yusuf Mücahit Çetinkaya[1] · İsmail Hakkı Toroslu[1] · Hasan Davulcu[2]

Received: 18 March 2020 / Revised: 7 June 2020 / Accepted: 12 June 2020

Abstract
Today, microblogging platforms like Twitter have become popular by spreading news and opinions that gather attention. Engaging interactions, such as likes, shares, and replies, between users are the key determinants of these platforms' news feed prioritization algorithms. These interactions attract people to ongoing debates and help inform and shape their opinions. Since being influential and attracting followers in these debates are considered as important, understanding the automation of these processes becomes critical in order to contribute positively. In this work, we aim to train a chatbot system that classifies tweets according to their positions, and it can also generate tweets related to a conversation. In this study, we test our system on a recently popular topic, namely the gun control debate in the U.S. Chatbots, are trained to tweet independently for their side and also reply meaningfully to a tweet from the opposite side. State-of-the-art architectures are tested to obtain a more accurate classification. We applied GloVe embedding model for representing tweets. Instead of using handcrafted features, long short-term memory (LSTM) neural network is applied to these embeddings to get more informative and equal-sized feature vectors. This model is trained to encode a tweet as a sequence of embeddings. Encoding is used for both message classification and generation tasks. LSTM sequence-to-sequence model is used to generate topical tweets and replies to tweets. We develop a new salience metric for measuring the relatedness of a generated message to a target tweet. Additionally, human evaluations are performed to measure the quality of the chatbot generated tweets according to their topic relevance and bias, and the quality of its replies to target tweets.

Keywords Twitter bot · Natural language processing · Tweet generation · Tweet classification · Sentiment analysis · Recurrent neural networks

1 Introduction

Twitter has become very popular since it is founded in 2006 due to its rapid information diffusion property. Today, users mostly use it for sharing and commenting about breaking news and events (Rosenstiel et al. 2015). Even it was not mainly designed for fostering interactions (Liu et al. 2010), just like on other social media platforms, Twitter users argue with supportive and opposing replies to tweets posted by other users. This interaction causes users to not only promote tweets that they agree with through likes and shares but also get into heated arguments and debates with others whom they disagree with. These debates cover a wide variety of topics, from sports to politics, economics to culture, covering all societal issues. Nowadays, any debate related to any popular topic can be easily found and participated on Twitter.

It is the nature of adversarial debates to argue for one's position and persuade one's opponents to adopt their perspectives. However, social media debates are different from the ones in real life. In our study, we had an opportunity to examine a large collection of tweets related to a highly polarizing and controversial issue. We found that majority of the tweeters are partisans who consume and produce content with only a one-sided leaning, and they blame the opponents

✉ İsmail Hakkı Toroslu
toroslu@ceng.metu.edu.tr

Yusuf Mücahit Çetinkaya
yusufc@ceng.metu.edu.tr

Hasan Davulcu
hdavulcu@asu.edu

[1] Department of Computer Engineering, Middle East Technical University, Ankara, Turkey

[2] School of Computing, Informatics, and Decision Systems Engineering, Arizona State University, Tempe, AZ, USA

Published online: 01 July 2020

for the problem and want them to find a solution. Moderate and bipartisan users who produce content with more tolerant leanings and make an effort to bridge the echo chambers are less valued in their networks. A recent study (Garimella et al. 2018) of more than 2.7 billion tweets between 2009 and 2016 confirms that Twitter users are exposed mainly to opinions that agree with their own, and partisan users enjoy higher appreciation measured by both their network centrality and content endorsement. The findings indicate a strong correlation between biases in the content people both produce and consume. In other words, echo chambers are very real on Twitter, and bridging them is a daunting task.

Bots that imitate partisans use social media echo chambers effectively as they arrange campaigns that exploit users' cognitive biases and cultural preferences with catchy slogans, hoaxes, fakes, and opinions which reverberate and spread as they are being shared by others in the chambers. Automated botnets now exist to exploit echo chambers and further divide and polarize communities (Varol et al. 2017). Research reveals malicious activities of such botnets during the U.S. Presidential Debate (Howard et al. 2016). Coordinated and synchronized botnets can now be programmed to amplify tweets matching certain partisan perspectives or from certain fringe users. In social media, reposting the same tweet multiple times is referred to as spamming. Even though effective methods exist for detecting and removing spamming botnets (Chavoshi et al. 2016), it remains challenging to do attribution to identify their operators, and they tend to be replenished rapidly due to the open nature of social networks.

Malicious bots are one of the nocuous entities in social networks. Although it is not the main focus of this study, we need to better understand the biases that partisan botnets exploit and counter them by naming and shaming, and by showing how they dupe unsuspecting groups by undermining their very values that they purport to defend. Social media companies made significant investments in human and automated systems to detect and remove problematic content with artificial intelligence algorithms and tens of thousands of new employees to help them patrol the onslaught of fake and malign content posted on their platforms. However, they have a content-moderation problem that is fundamentally beyond the scale that they know how to deal with.

To the best of our knowledge, this study makes a step towards developing an intelligent chatbot that aims to bridge opposing partisan echo chambers by diversifying opinions and assisting in the emergence of a consensus through civil discourse. It aims to achieve this moderation by replying to tweets and starting new conversations by linking not-too-distant perspectives and users on both sides of adversarial and polarizing debates. Earlier, Rakshit et al. (2017) developed Debbie as an arguing chatbot; however, Debbie uses a retrieval-based approach repeating earlier messages that may look like spam, and it is limited to predefined responses as opposed to learning-based approaches that we present here.

The rest of the paper is organized as follows. In the next section, we review related works. In Sect. 3, we introduce the proposed methodology. Section 4 describes the dataset and preprocessing details. Section 5 describes our classification model. In Sect. 6, we present the details of the tweet generation process. In Sect. 7, we present the reply generator model, a new relatedness metric, and evaluations. In Sect. 8, we conclude the paper and discuss future work.

2 Related work

There have been various researches on developing software that can mimic human beings in conversations for a long while. Although researches in this domain started in the 1960s with a project called ELIZA, which is a text-based chatbot developed in MIT AI Laboratory (Weizenbaum 1966), chatbots have drawn significant attention in recent years due to the advancements in the technology and social media platforms. Earlier systems perform the tasks with handcrafted rules (Bradeško and Mladenić 2012). As distinct from the earlier systems, contemporary models rely on data-driven approaches (Tao et al. 2019).

The systems can be categorized in different perspectives, like closed domain (Mondal et al. 2018; Xu et al. 2017) or open domain (Higashinaka et al. 2014; Yan 2018), conversation type (e.g., short-text dialogue (Su et al. 2017; Li et al. 2016), question answering (Quarteroni and Manandhar 2007; Wang and Jiang 2017), etc.), response generation type (e.g., retrieval-based Wu et al. 2017, 2018, generation-based Shang et al. 2015; Yin et al. 2017, or hybrid Qiu et al. 2017; Tammewar et al. 2018), etc.

Closed-domain chatbots are developed concerning for specific tasks. The dialogues turn around a particular subject. Cui et al. (2017) proposed a browser add-on extension called SuperAgent, which mimics online customer support and answers user's questions about e-commerce products according to the product webpage. Some chat applications like Facebook Messenger, Telegram, Whatsapp provide APIs for sending and receiving messages. Holotescu (2016) is an online course recommender system based on the user's social media profile and interests serving on Facebook Messenger as a chatbot. Task-oriented chatbots provide service for a particular domain, where usually do not support open-ended conversations, and the user is aware that the responder is a bot. Some chatbots are deployed for quick and specific tasks such as checking the weather, organizing meetings, ordering food, or booking a flight, while others support more long-term relationships and activities such as a charity or civic engagement, work, fitness, and personal health (Brandtzæg and Følstad 2018). These chatbots can interpret

a limited subset of queries or offer options to the user to select among them (Brixey et al. 2017; Parthornratt et al. 2018; Mostaço et al. 2018; Roca et al. 2020).

Open-domain chatbots are built to talk without conversation limitation (Hussain et al. 2019). In contrast to traditional task-oriented chatbots, the primary goal of open-ended dialog chatbots' is to establish emotional connections to humans rather than completing specific tasks, and they often try to mimic human conversations (Huang et al. 2020).

Retrieval-based models bring the closest source sentence to the user request from their candidate database according to a ranking mechanism and return the predefined response from the repository (Chen et al. 2017). On the other hand, generative models employ their tasks by generating responses based on the user request. For this purpose, they have a mechanism that the language model is encoded inside. Generative models often adapt the sequence-to-sequence recurrent neural network (seq2seq-RNN) technique used in neural machine translation (Bahdanau et al. 2015), where the model encodes the source sentence into a fixed-length vector to decode the translation from it. In dialogue generation tasks, the encoder is fed with the request, and indented response is given to the decoder as the translation (Lei et al. 2018). They can capture the meaning and generate relevant responses to the unobserved requests. Radford et al. (2019) construct an auto-regressive feed-forward model instead of seq2seq-RNN as a language model using Common Crawl as a dataset and generate sentences with predicting next word.

Generative models are more prone to make mistakes than the retrieval-based models since they are generating the sentences from scratch (Ramesh et al. 2017). The mistakes might be grammatical or semantical caused by losing the context, especially for long sentences. Attention mechanisms that are developed to solve the long-term dependency issue are a widely used add-on to the seq2seq models (Bahdanau et al. 2015; Luong et al. 2015).

As mentioned above, both retrieval-based and generation-based models should be able to comprehend the user's request somehow. In addition to dialogue generation, interpretation of the sentences might be used for different purposes, such as classification (Cui et al. 2020; Ertugrul and Karagoz 2018), event detection (Goswami and Kumar 2016; Ertugrul et al. 2017), and anomaly detection (Mahapatra et al. 2012; Ebrahimi 2016).

Generative adversarial networks (GANs) get popular after its success on computer vision tasks (Radford et al. 2016; Antipov et al. 2017; Karras et al. 2019). The complete model consists of two models, generator and discriminator, to generate new data by looking at the probability distribution of the given training set (Goodfellow 2016). While the generator produces samples similar to the real data distribution, discriminator tries to distinguish a generated sample from a real one. It is also used in recent works in the NLP domain, such as chatbots (Kim et al. 2019), neural machine translation (Yang et al. 2018), and image captioning (Chen et al. 2019). Similar to traditional dialogue models, GAN-based models on text generation can be categorized as retrieval-based and generative-based models. In Wang et al. (2017), the model produces text with a retrieval-based approach that generator tries to learn document relevance distribution to generate document pairs with the correct ranking, while the discriminator tries to distinguish such generated document pairs from real document pairs. Li et al. (2017) propose a GAN model for dialogue generation with a seq2seq model for the generator part.

In the past decade, big companies have made investments on intelligent personal assistants (IPAs), such as Apple Siri, Microsoft's Cortana, Google Assistant, Facebook M, and Amazon's Alexa (Shum et al. 2018). They can be considered as a type of chatbot. They understand what user requests and respond with appropriate answers accordingly. Additionally, IPAs proactively anticipate user needs and provide in-time assistance, such as reminding of an upcoming event or recommending a useful service without receiving explicit requests from the user (Sarikaya 2017).

Microsoft's chatbot XiaoIce has been designed as a 19-year-old female persona, with strong language ability, visual awareness, and over 180 skills (Shum et al. 2018). Microsoft released Tay in 2016. It behaves like an 18–24-year-old American woman and has conversations over Twitter. XiaoIce and Tay can expand their knowledge during the conversations. This skill made Tay learn offensive language and content that causes Microsoft to take the chatbot down (Neff and Nagy 2016).

3 Methodology

In this study, we use deep neural networks, namely RNNs, for addressing the issues mentioned above since they have remarkable success on sequential data. Long short-term memory (LSTM) is a specific type of RNN that fits time series data. It has some special units to remember previous values over predefined time intervals, which makes it able to keep the context until the end (Hochreiter and Schmidhuber 1997). Neurons inside the LSTM model are called memory cells. It has variety of application areas, such as classification (Karim et al. 2018), generation (Wen et al. 2015), and neural machine translation (Sutskever et al. 2014).

Since Twitter has no strict rules for texts, tweets might be grammatically incorrect and contain misspelled words or multifarious smileys. They need to be projected onto the same plane to reduce the probable errors. We have preprocessed retrieved tweets with some rules, such as lowering letters, using tags instead of URLs, mentions, hashtags, and multiple characters reduced to 2. It also reduced the

vocabulary size conspicuously, which avoids the curse of dimensionality.

Computers must get symbols as numeric values. Since representing each word with a bag-of-words approach ends up with huge sparse matrices, it is essential to reduce the vectors' dimensions to the considerable size. Topic modeling is a machine learning technique that tries to determine the latent topics from the corpus with statistical information of word-document mappings. The documents that are considered to be in the same latent topic are close in the vector space. Latent semantic analysis (LSA) is a method for representing each word with vectors in an unsupervised manner, where the bag-of-words vectors are reduced with singular value decomposition (Dumais et al. 1988). Latent Dirichlet allocation is a Bayesian approach that tries to model text corpora and represent each latent topic with a combination of words with different coefficients (Blei et al. 2003). Representing the words with vectors might be established using that word's coefficients of each topic.

Word embedding is a similar approach to symbolize the words with vectors. There are different embedding models for representation of words like GloVe (Pennington et al. 2014), word2vec (Mikolov et al. 2013), fastText (Bojanowski et al. 2017), etc. In the embedding space, each word represented with a coordinate related to its context. Therefore, semantically close sentences are located in this space as close as they can. Word embedding technique is used in different domains for different purposes. Demirel et al. (2019) propose a template-based image captioning where it uses word2vec embeddings to represent class names of images while obtaining similar classes to retrieve a template and GloVe embeddings for the words in the template. Peng and Jiang (2016) try to predict stock price movements based on financial news and use word2vec embedding for expanding keyword list that is started with manually selected seed keywords. Chollampatt and Ng (2018) introduce a neural network model for detecting and correcting grammatical errors where words are represented with fastText word embeddings.

There are pre-trained models shared on the study webpages. The word2vec model is pre-trained with Google News corpus with 3 billion tokens, which ends up with 3 million 300-dimensional English word vectors. The fastText model consists of 2 million word vectors trained on Common Crawl with 600 billion tokens (Mikolov et al. 2018). GloVe vectors are trained with Twitter data containing 2 billion tweets, with a total of 1.2 million unique words of 27 billion tokens. Our dataset is a relatively small corpus as compared to referent models; hence, we proceed with the pre-trained models. Since the dataset we are working on is from Twitter, it is better to use a model that can comprehend Twitter keywords, slangs, and abbreviations. Table 1 shows the closest seven words to some words from Twitter terminology with comparing pre-trained word2vec, fastText, and GloVe models. The pre-trained GloVe model returns semantically close words in the Twitter world, while others behave as they are used in regular language. Therefore, we decided to use the GloVe model that is trained with a Twitter dataset.

GloVe, global vectors for word representation, is an unsupervised model for representing the words using co-occurrence probabilities retrieved from large corpora. The statistics of word occurrences in a corpus is the primary source of all unsupervised methods for learning word representations (Pennington et al. 2014). The interrelationship between words is discovered using this co-occurrence data. The study focused on generating the meaning using this information and representing it with a vector.

For the first LSTM model that is responsible for predicting the side of the tweet, the input is fed from the tweet as a sequence of embedding vectors. Each word is given one by one, and the state stored in the cell is updated according to previous embeddings during forward propagation to the network. At the end of the sequence, the cell encapsulates the representation of the whole tweet that is encoded by the network. It is then connected to the fully connected layer, which classifies the tweet. The model predicts the probabilities of each label that the tweet may belong to. For example, the tweet "keeping and bearing arms is our

Table 1 Top seven closest words to *"retweet, follow, mention, dm"* words from pre-trained word2vec, fastText, and GloVe embedding models. The models are trained with different corpora

retweet			follow			mention			dm		
word2vec	fastText	GloVe	word2vec	fastText	GloVe	word2vec	fastText	GloVe	word2vec	fastText	GloVe
Tweet	Retweet	rt	Follow	Follw	Following	Mentioning	Metion	Mensyen	sc	DM	kik
Twitter	Retweeting	Tweet	Followed	Folllow	Back	Mentions	Mentioning	Mentions	ro	dm.	Reply
Tweet	ReTweet	fav	Follows	Folow	Followback	Mentioned	Meantion	Reply	dr	dms	Email
Twitter	Retweet	nrt	Adhere	Followed	Please	Allude	Mention	Twit	te	DM.	Inbox
Facebook	Retweets	Mention	Abide	Follow	Followers	Alluded	Memtion	Twitt	va	Dm	Skype
Tweets	Retweet	Retweets	Following	Foolow	Follows	Forget	Mentioned	Metion	ot	hc	fb
Tweets	Retweeting	Follow	Adhered	Follwo	Retweet	Mention	Meniton	Bales	uk	Dms	Mention

constitutional right" is converted to a sequence of embeddings, ⟨[0.33224, − 0.18878, … , 0.22622, − 0.06764], … , [0.077965, − 0.24248, … , 0.15298, 0.53293]⟩. The memory state of the LSTM at the end is used for classification by connecting fully to the output nodes. The resulting output for the tweet is $o_{right-leaning} = 0.82$ and $o_{left-leaning} = 0.18$, where the prediction of the label is right-leaning with 82% probability.

The tweet generation model has a marginal difference compared to label prediction. It predicts the next symbol with the given sequence of symbols instead of the label. The symbol may differ for varied tasks; however, we use the word embedding as a symbol. Using character as a symbol makes the model more prone to losing the context. Like the prediction task, we fed the network with a series of word embeddings by hiding the last one. The hidden embedding is the targeted output that the network should assert. This technique is called as next-word prediction or language modeling. While predicting the upcoming word, "arms," with the preceding word sequence of "keeping and bearing," the model parameters are updated accordingly. It processes a single word at a time. The fundamental statistical properties of the language are learned with this approach. The prime text is the initial input for the model to complete the rest. Primes are the single or multiple words selected from the pre-determined set.

Only next-word prediction is not enough for getting interacted with other users. The bot should give responses to other users. The responder model should consume the whole text to get the meaning and generate a reply. This procedure is similar to the NLP task called neural machine translation. In such models, LSTM consumes the sequence and produces the representation as we do in the tweet classification. This representation is used to bring the replies out. More clearly, the model consists of two separate parts; encoder and decoder. The encoder is analogous to the recurrent part of the classification task. However, instead of connecting the cell to the fully connected layer, it is connected to the decoder. For example, the model takes the embedding sequence of the tweet "fox is reporting actual news ? !" and the memory state of the cell encapsulates the context into a vector after completing the tweet. The decoder is comparable with the one in the generation model, but it does not get a prime text. Instead, it takes the source tweet's encoding and generates a response for it with next-word prediction. The predicted words are compared with the sample reply, "as always dude," to the example source tweet for computing the gradient. This technique has been quite successful in automated translations (Cho et al. 2014).

Generated tweets might have some deficiencies. Therefore, they need to be checked concerning the semantic and the relatedness. In particular, a bot to be a right-leaning account should not post a tweet supporting gun control. If the semantic of the tweet does not reflect the view correctly, it is not posted. Similarly, the generated reply to the target tweet is required to be related somehow. A response generated for "how much money did you get from the nra ?" should not be "teachers should own firearms." Generating opposing view tweets or unrelated tweets is possible with generation-based models. An automated filtration of such tweets with quantitative metrics is crucial. The bot ignores and does not post the generated tweets that do not exceed a threshold for both semantic metric and relatedness metric.

Figure 1 visualizes the flow of how the bot carries out the fundamental operations on Twitter. The retweeting module classifies the tweets before retweeting them. If the tweet reflects the view of the bot with a certain probability, it is retweeted. The tweeting module contains the LSTM model for tweet generation. It generates tweets starting with given prime and then sends it to the classifier. Produced tweet needs to satisfy the threshold criteria from the semantic check. If it exceeds the threshold, it is posted. For example, the generated tweet, from a left-leaning bot, "protect constitutional rights to protect our family from murderers" reflects the right-leanings' thoughts with 92% probability. Consequently, the bot neglects the generated tweet and does not post. The responding module is responsible for replying to the opposite view tweets. The target tweet is classified, and if the tweet is dissident to the view of the bot, the reply generation model is fed with the tweet. The generated reply is checked according to semantic and relatedness, respectively. If it passes both filtration mechanisms, it is posted.

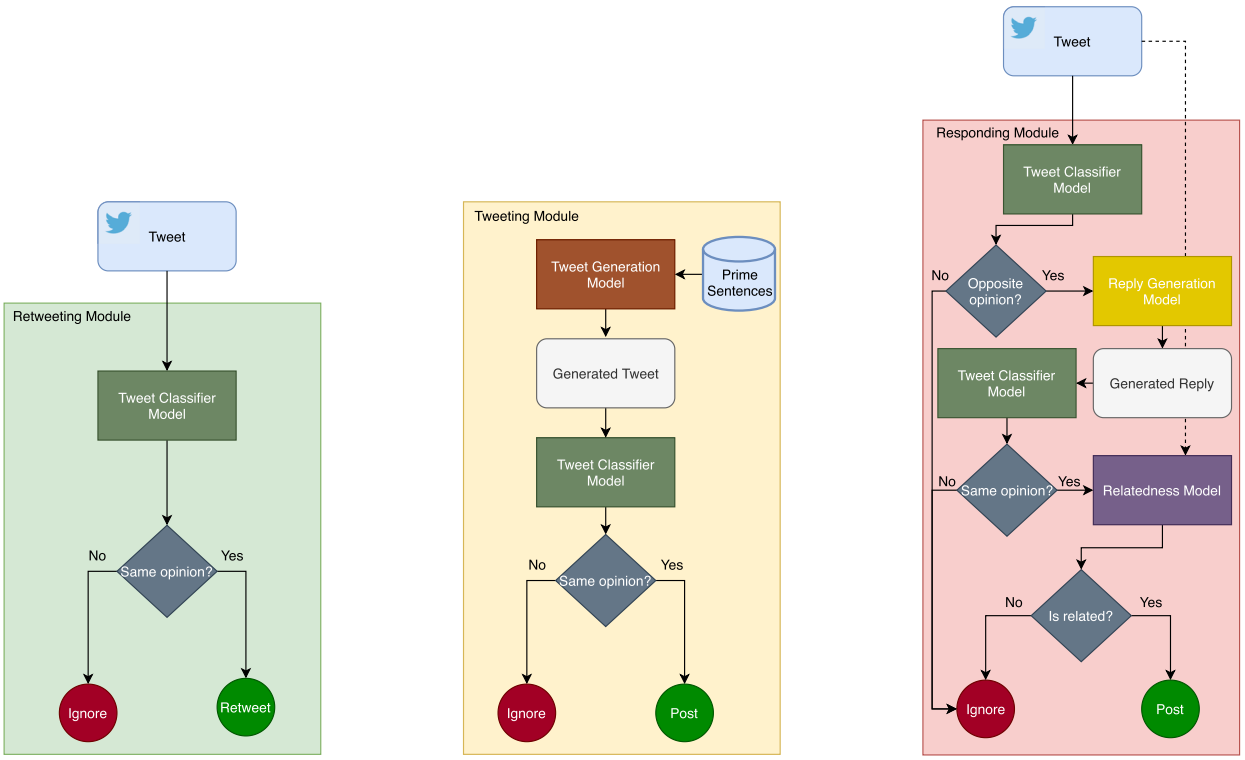
4 Dataset

4.1 Details

The dataset that we work on in this study contains nearly 24 million tweets in English about gun control in the USA. There are two sides of the discussion, pro-gun, and anti-gun. Tweets are posted by 3,282,592 individual accounts. Users are labeled instead of tweets. The label of the tweet is accepted the same with the label of the owner. 293,046 of the users are right-leaning, while others are left-leaning. Right-leanings are considered as pro-gun accounts. Almost 5 million tweets belong to right-leanings. Although the dataset is skewed, we sample equally from each side during our experiments. The words used in both sides of the debate are highly overlapped, as shown in Fig. 2.

4.2 Preprocessing

Since Twitter limits the length of tweets, people use abbreviations, slangs, etc. This makes processing tweets harder. Additionally, emojis do not have standards. Any sequence of characters that looks like an entity is used as emoji.



(a) Flow diagram of retweeting process. **(b)** Flow diagram of tweeting process. **(c)** Flow diagram of replying process.

Fig. 1 Proposed architecture of the bot consists of three independent modules. It has the ability of retweeting, tweeting, and generating replies to the dissident tweets. If the generated content has not intended quality, it is not posted

Therefore, we have to process tweets before using them in NLP tasks. The steps described for preprocessing in GloVe webpage are followed in general. All letters are converted to lowercase to get rid of the variation of the same words. Our bot does not have a module that can crawl a webpage and analyze the sentiment. Hence, the hyperlinks are not needed, and URLs are replaced with ⟨*url*⟩ tag. Similarly, ⟨*user*⟩ tag is inserted for any user mention inside the tweets. Hashtags might be used in the tweets while forming the meaning, but they are mostly used to support the idea. ⟨*hashtag*⟩ token is used for any hashtag. We have replaced all numbers with ⟨*number*⟩ tag because treating each number as a separate word is not possible. Punctuations other than {!, ., ?, %} are removed. Simple emojis are replaced with the definitive tags, such as ⟨*smile*⟩,⟨*lolface*⟩ ,⟨*neutralface*⟩, and ⟨*sadface*⟩. Emoticons, which are emojis with Unicode characters, are purged. Lastly, we have replaced consecutively repeated characters that occur more than twice with two characters. Table 2 includes before and after preprocessing applied of sample tweets.

5 Political view estimation of the Tweet

Experiments showed that the trained model learned the semantic of tweets about gun control well. While the models' prediction probabilities are getting higher, accuracy is also increasing. This correlation implies that the model has the ability of a proper generalization of tweets. Even it cannot classify all tweets correctly, it is sufficient for a Twitter bot to interpret a substantial number of tweets accurately. Therefore, it is the right approach for relying on the tweets where the bot's prediction probability is above a threshold, such as 80%.

5.1 Classification problem

Right-leaning people feel that keeping and bearing arms is an essential right for citizens. Left-leanings are opposed to this idea and think that it is the main reason behind most

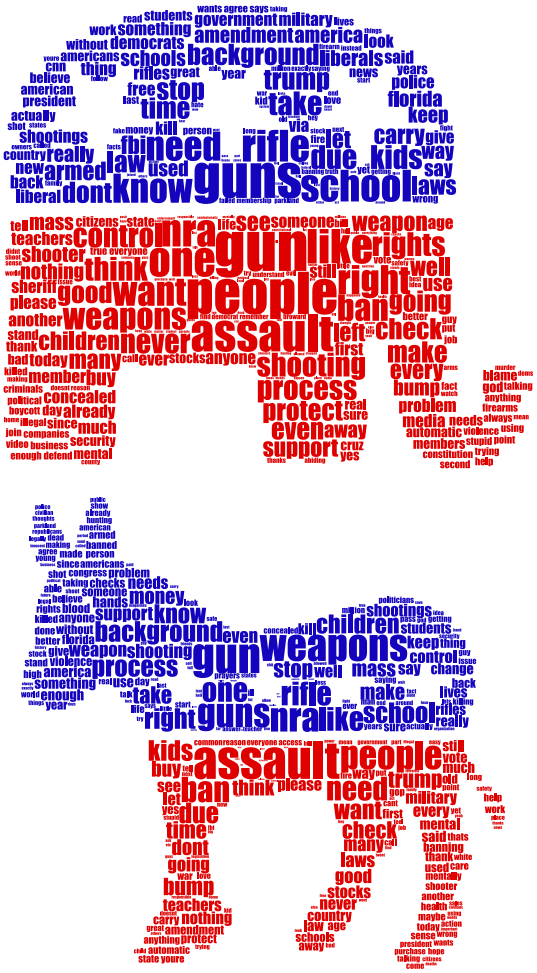


Fig. 2 Word clouds that represents the top 350 words used in the tweets of both sides of the debate separately. The sides are represented with the symbols of the Republican Party (elephant) and the Democratic Party (donkey) of the USA, where their supporters are mostly labeled as right-leaning and left-leaning, respectively

of the deaths. In reality, it is not easy to draw a strict line between these opinions. There are various point of views, such as supporting owning only handguns but not rifles, demanding that all guns should be prohibited, and making mandatory to do a detailed background check before purchasing a gun. Since grouping each view eventually ends up to the number equal to the individuals exist, sides are labeled as right and left.

5.2 Encoding the Tweets

In this section, we explain how we feed the LSTM to get the related encoding. The main aim of the network is moving sentences with natural language into the computer encoded space where semantically close tweets are close to each other. The dataset is represented in this section as $D = (S, L)$, where $S = \langle s_1, s_2, s_3, \ldots, s_N \rangle$ and $L = \langle l_1, l_2, l_3, \ldots, l_N \rangle$ stand for the tweets and the labels. Tweets are tokenized to get sequences of words. Tweets, s_i, comprise of tokens denoted as $\langle t_1, t_2, t_3, \ldots, t_m \rangle$ where m is the length of the tweet. Each token is represented with a GloVe embedding vector. Embedding vector representation of the tweet, s_i, is expressed as $\langle v_1, v_2, v_3, \ldots, v_m \rangle$. Figure 3 visualizes the lookup method while generating the vector representation of the tweet. Computed encoding of the tweet, s_i, is denoted as e_i.

Since pre-trained GloVe vectors are used, the contexts of the words are already preserved in their vector representations according to their co-occurrence frequencies. The model is not responsible for extracting the relationship between words. Instead, it captures the meaning according to the arrangement of these words. LSTM cells keep a state inside, which is calculated with given sequences. The hidden state is updated when every vector, v_j, is sent to the LSTM cell. The state depends on the previous vectors given, $\langle v_1, v_2, v_3, \ldots, v_m \rangle$. In the end, the state, e_i, is the feature itself that is used for classification. The resulting state is important since the meaning can be captured only when the sentence is completed. Therefore, we are not interested in intermediate states.

We want the model to be more robust to unseen data. Therefore, the dropout technique is used in LSTM cells. It randomly chooses nodes and removes them for the current iteration. At each iteration, different neurons might

Table 2 Sample tweets after preprocessing is applied

@RealDavidTafoya @charssocialism @shanragirl @moledwh79 @theogeo14 @KyleKashuv @kjcolvin611@davidhogg111 1.) The NRA does more than abortions. 2.) Stay out of women's health. ur bodies, our choice.
⟨user⟩ ⟨user⟩ ⟨user⟩ ⟨user⟩ ⟨user⟩ ⟨user⟩ ⟨user⟩ ⟨user⟩ ⟨number⟩. the nra does more than abortions. ⟨number⟩. stay out of womens health. ur bodies our choice.
@sweetestswim @LynseyDee2 @denise_kandra @TobyasStorm @gilbert87 @davidhogg111 @NRA And theClintons received hundreds of millions of directly or through their foundation. hat does that make her and the Dems?
⟨user⟩ ⟨user⟩ ⟨user⟩ ⟨user⟩ ⟨user⟩ ⟨user⟩ ⟨user⟩ and the clintons received hundreds of millions of directly or through their foundation. hat does that make her and the dems ?
83% of Democrats and 54% of Republicans support raising the minimum gun purchase age to 21. So why isn't anything being done? The @ NRA is bribing politicians, apparently sometimes even with Russian money. WAKE UP AMERICA!
⟨number⟩ % of democrats and ⟨number⟩ % of republicans support raising the minimum gun purchase age to ⟨number⟩. so why is not anything being done ? the ⟨user⟩ is bribing politicians apparently sometimes even with russian money. wake up america !

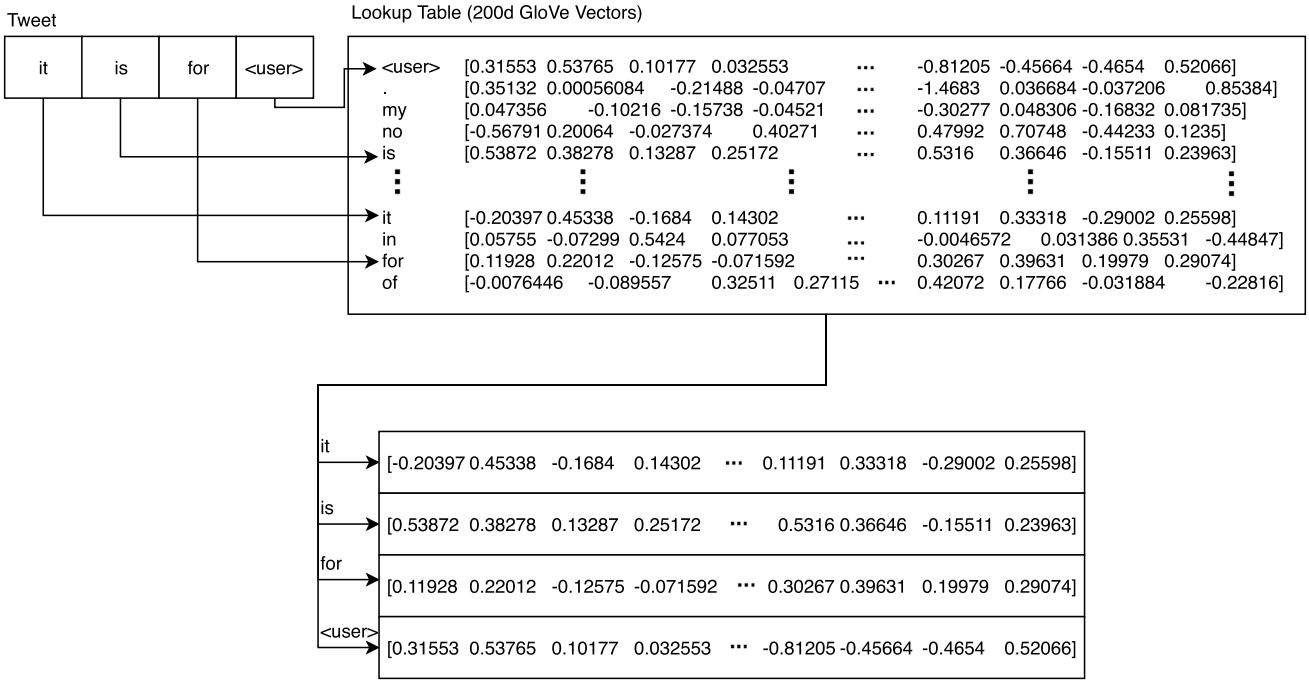


Fig. 3 Lookup method for generating tweet representations

be chosen. A hyper-parameter is used for detecting how many nodes to drop out. This method allows nodes to learn their weights independently. Otherwise, they might work together to represent the input during training, making the model memorize training data. Each neuron is specialized in extracting deep features with the coordination of random neurons.

5.3 Classifying the encoding

Figure 4 visualizes the architecture of the model. The output of the LSTM, e, is connected to the fully connected layer for classification. There are output nodes as many as unique labels, which are two in our dataset. The size of the encoding may vary according to the model, and it is denoted as D in this subsection. Every dimension of the encoding, e_j, is connected to these outputs with trainable weights, w_{jk}, where k stands for the label node. An additional bias term is added to the multiplication. Output of the label node o_k can be formulated as $\sum_{j=1}^{D} w_{jk} e_j + b$. The numeric output that is fired from each label node is calculated separately and is called logits. These values are not compressed into a range without an activation function. Since being right-leaning and being left-leaning are mutually exclusive classes, we need to convert these logits into probabilities. Softmax activation function, $softmax(o_k) = \frac{e^{o_k}}{\sum_t e^{o_t}}$, is applied then to o_k for adding nonlinearity and getting probabilities of each label instead of logits.

We used cross-entropy loss since the outputs of label nodes are the probabilities where tweet may belong to. In addition to

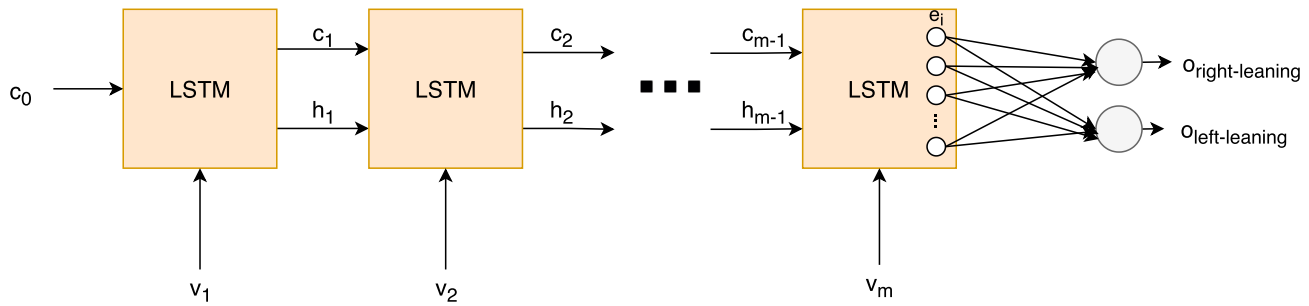


Fig. 4 Unfolded version of the LSTM model. At each time step, the model state is updated according to v, c, and h corresponding to current input, cell state, and hidden state. In the end, the state, e_i, is the feature itself that is connected to the fully connected layer for classification

cross-entropy loss, we added L2 regularization term for avoiding the over-fitting problem. Adding the regularization term, the sum of the squares of the weights with a coefficient, to the loss causes the weight's values to decrease, which is accepted as a simple model. Therefore, it will prevent the model from memorizing training data. L2 regularization is also known as weight decay since it adjusts the weights close to zero.

5.4 Experiments

In the experiments, we used 200-dimensional GloVe vectors, which are pre-trained with Twitter data. 400,000 tweets are used on both sides. If a word is absent in this data, we used a uniformly random vector with the values in range $(-0.25, 0.25)$. The size of the LSTM cells is 128. Drop-out probability and L2 regularization coefficient λ are 0.5 and 3, respectively. The initial learning rate for updating the weights according to the gradient is chosen as 10^{-4}. Ruder (2016) compares multiple gradient descent optimizers for adjusting weights, concluding that choosing an adaptive learning rate method is better where ADAM might be the best overall choice. Therefore, we use ADAM as an optimizer for the model.

Logistic regression is used as a baseline for the proposed model. Tweets are represented with the bag-of-words technique with three different ways: occurrence, count, and tf-idf.

We have applied k-fold cross-validation to estimate the generalization performance of the models in a consistent way. The number of folds, k, is decided as 7.

Table 3 shows the model performances in detail. The accuracy we obtain is 84.81% on average of sevenfold, which is the best score among the models for each metric. Besides, just like real Twitter users, our bot should not interact with all accounts concerning casting doubt on it. Therefore, we come up with the idea of filtering these tweets to detect which tweets should be taken into account or not. This filtration is done with a threshold of prediction probability. We examine the performance of our classifier on these filtered tweets. It showed that our model predictions are more accurate on these tweets.

As shown in Table 4, the tweets that our model predicts the label with the probability above 85%, cover 50.36% of all test set on average. However, it dramatically boosts the performance up to 95.36%. Some tweets do not show an opinion specific to the side. Table 5 contains the tweets that our model predicted the label with low probabilities.

6 Tweet generation in the context of a debate

Posting predefined tweets from multiple accounts is a wrong approach since it is easy to be disclosed by other users. Therefore, generating random but meaningful texts is a must for a smart bot. As mentioned in Sect. 5, the arrangement of the words is the main feature of forming the semantic. While generating text from scratch, the computer starts with

Table 3 Classifier performance results comparing to the baselines with cross-validation with sevenfold

	Accuracy		$\text{Precision}_{\text{left}}$		$\text{Recall}_{\text{left}}$		$\text{Precision}_{\text{right}}$		$\text{Recall}_{\text{right}}$	
	Mean	Std.	Mean	Std.	Mean	Std.	Mean	Std.	Mean	Std.
LR (binary counts)	81.65	0.108	83.70	0.126	80.41	0.161	79.61	0.210	83.00	0.184
LR (multiple counts)	79.20	0.135	79.12	0.213	79.25	0.342	79.28	0.364	79.15	0.165
LR (Tf-Idf)	79.79	0.258	80.39	0.261	79.43	0.339	79.18	0.345	80.15	0.294
Proposed model	**84.81**	0.146	**89.28**	0.577	**81.95**	0.292	**80.33**	0.556	**88.23**	0.472

The best results are shown in bold

Table 4 Detailed performance results of the proposed model and the ratio of tweets that classifier has prediction probability above different filter threshold values to the whole test set with cross-validation with sevenfold

τ	Tweet ratio		Accuracy		F-Measure	
	Mean	Std.	Mean	Std.	Mean	Std.
0.50	100.00	0.0	84.81	0.146	85.46	0.192
0.55	95.02	0.316	86.51	0.138	87.10	0.155
0.60	89.84	0.653	88.08	0.192	88.57	0.198
0.65	84.28	0.985	89.60	0.256	89.94	0.230
0.70	78.13	1.231	91.05	0.334	91.20	0.290
0.75	71.27	1.407	92.47	0.311	92.37	0.247
0.80	63.05	1.378	93.82	0.250	93.33	0.178
0.85	50.36	1.259	**95.36**	0.147	**93.78**	0.362

The best results are shown in bold

Table 5 Sample tweets that the model predicts the label with low probabilities

Tweet	P_{left}	P_{right}	True label
⟨user⟩ regular normal people know this ⟨url⟩	51.70	48.30	Right
this is peak crazy america ⟨url⟩	51.87	48.13	Left
⟨user⟩ a background check on potential political candidates is not a bad idea either	48.07	51.93	Left
warning you can not unwatch this. ⟨url⟩	52.23	47.77	Left
cant do anything about social posts sry this is america still. ⟨url⟩	46.21	53.79	Right
this whole video is a joke these people are so ignorant ⟨url⟩	54.63	45.37	Left

a given prime word and adds a new one that is more probable to follow it until the end.

In this section, we need to use a quite different LSTM model than the one adopted in the classifying task. Instead of predicting the label of the full tweet, the aim is to predict the next word in accordance with the current state depending on preceding words up to a point. During the model determines each word, the state is updated, and the meaning appears. GloVe embeddings are used for representing the words. Two models are imitating two sides of the debate. One is trained with the tweets of the left-leaning accounts, and the other is with the right-leanings. Generated tweets seem admissible in pursuant of human evaluators.

6.1 Designing model for next-word prediction

We bring each tweet in the dataset, $s_i = \langle t_{i1}, t_{i2}, t_{i3}, \ldots, t_{im} \rangle$, to same length, M, by padding them with special tag denoting the end, $\langle end \rangle$. For example, the tweet "guns must be banned" becomes a sequence ⟨guns, must, be, banned, ⟨end⟩,⟨end⟩, … ,⟨end⟩⟩ with the length of M. In this manner, the model learns when and how to conclude the tweet. LSTM decides to put an ending token when it completes the tweet generation. Otherwise, it computes next words up to predefined sequence length without a stop. In the early epochs, the model understands that when $\langle end \rangle$ token appears, it should be continuously followed with other $\langle end \rangle$ tokens up to the sequence length.

We define vocabulary as a tweet word set with a size of N. The output layer consists of N nodes, each representing one word in the vocabulary. The vocabulary is generated from the training set. Similar to label prediction, the model computes the probability of each word.

Since we aim to construct a model that can form a sentence by predicting the next word, targeted outputs are prepared with the inputs itself as visualized in Fig. 5. Tweets are one word shifted to obtain the inputs and the labels. For instance, if the tweet, s_i, is *"the truth will set you free"*, then input and output are, respectively; $x_i = [\}\}the'', \}\}truth'', \}\}will'', \}\}set'', \}\}you'']$ and $y_i = [\}\}truth'', \}\}will'', \}\}set'', \}\}you'', \}\}free'']$. While moving on the sequence in time, the model slides over the input and the true output lists. The essential part is that the values in x_i list are GloVe vectors. However, the ones in the y_i are one-hot vectors where the cell indexed with the corresponding word is one.

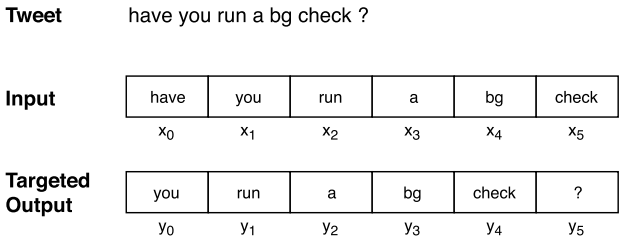


Fig. 5 A sample from training set prepared for next-word prediction task

Word prediction is done similar to the label prediction in Sect. 5. However, this time intermediate states, h_j, represent the intermediate words. The state is connected to a fully connected layer for estimating the most probable word. For making each value refer to a probability, the softmax function is applied to these logits. For optimizing the model, we use sequence loss, which is a cross-entropy loss. The difference is that it is calculated over a sequence with given weights. The mistakes done at the beginning of the sentence are as important as the ones done at the end. Therefore, we set all the weights equal. In total, it sums up the cross-entropy loss of overall tokens.

6.2 Constructing Tweets using the language model

The sampling phase is done after training is completed. The methodology followed for sampling significantly affects the outcome. Generated tweets may be monotone if they always trace the same path. In this subsection, we cover how to select words to arrange a sentence that reflects the opinion using the trained model. The aim is to generate unique tweets that look like a real human post. Hence, there is no absolute correctness for the outputs.

The model produces samples with a given prime word sequence. Results are more promising when the prime is selected from observed tweets in the training set rather than

randomly. As an instance, when a frequent expression "these are" is selected as a prime, a left-winger bot generates a tweet like "these are the people working for gun manufacturer supporters not republican willing and allows nra and politicians to sell their toys so their lives are yours !." Conversely, an unseen phrase in the dataset "he is the biggest" is followed up as "he is the biggest definition of a rifle paid wayne hair" by the same bot, which does not sense. However, unseen primes might also bring proper sentences out. For example, a word sequence from a song, "blow a kiss fire," does not exist in the dataset but is completed with a right-leaning bot as follows; "blow a kiss fire a gun with an assault weapons ban." Prime can be imagined as the preparation of LSTM. The state inside the cell needs to be initiated with a value. The model is incapable of forming a sequence with an empty state. When the state is ready for prediction, it completes the rest of the tweet.

We designed the model to estimate the most probable following word, up to the end. Fully connected layer computes the probability of each word to appear next with the current state of the cell. The word with the highest probability is accepted and appended to the sentence. We feed the model with the embedding of the chosen word to determine the following state, as shown in Fig. 6. This process repeats until the model outputs the ⟨*end*⟩ tag. Generated tweets seem very similar to the real ones. However, this approach always comes up with the same result for the same prime. Additionally, choosing the most probable word at each step may end up with a local minimum. Specifically, choosing the best word for a step might congest the meaning for the whole sentence.

We decided to add some randomness in choosing the word to get unique tweets. However, selecting the next word in an entirely random manner disrupts language modeling. Therefore, we sort the words by their probabilities and choose one of them randomly where there are more chances for higher probability words. This method increases the originality of the generated tweets. Besides, the sequence lasts longer because we force the model not to signal ⟨*end*⟩ tag early. Nonetheless, this situation leads the model to produce tweets with nonsense from time to time.

As a result, we go on with a heuristic for choosing the next word. The beam search is applied to the prediction probabilities. As demonstrated in Fig. 7, it expands given the number of possible branches for estimating the most promising sample. At each time step, non-promising alternatives are pruned from the hypotheses space. The sum of negative logarithm of all chosen words' probabilities in the sentence is used as a scoring value. Since the addition of the logarithm gives the multiplication of the values inside, the score represents the multiplication of probabilities of each prediction. Therefore, predicting a single word with high probability is not enough itself to accept a generated tweet to have a high score. It prevents exploring the wrong path by making one wrong prediction. Instead, further levels are considered. We observed that the beam search method produces more stable random sentences than picking the highest probability word, selecting a random word, and choosing a random word with the probability distribution.

6.3 Experiments

For representing the words, 200-dimensional GloVe vectors are used. The words that do not exist in the pre-trained GloVe model are represented with special tag ⟨*unk*⟩. There are two layers in LSTM. The size of the cells is 256. The

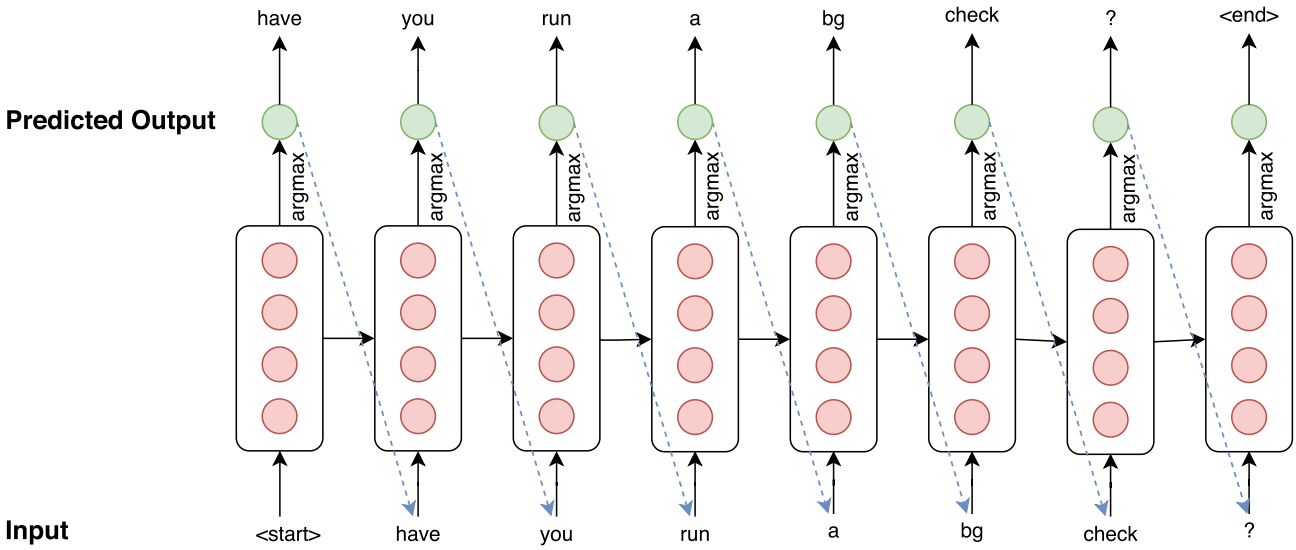


Fig. 6 Tweet generation using LSTM model with next-word prediction

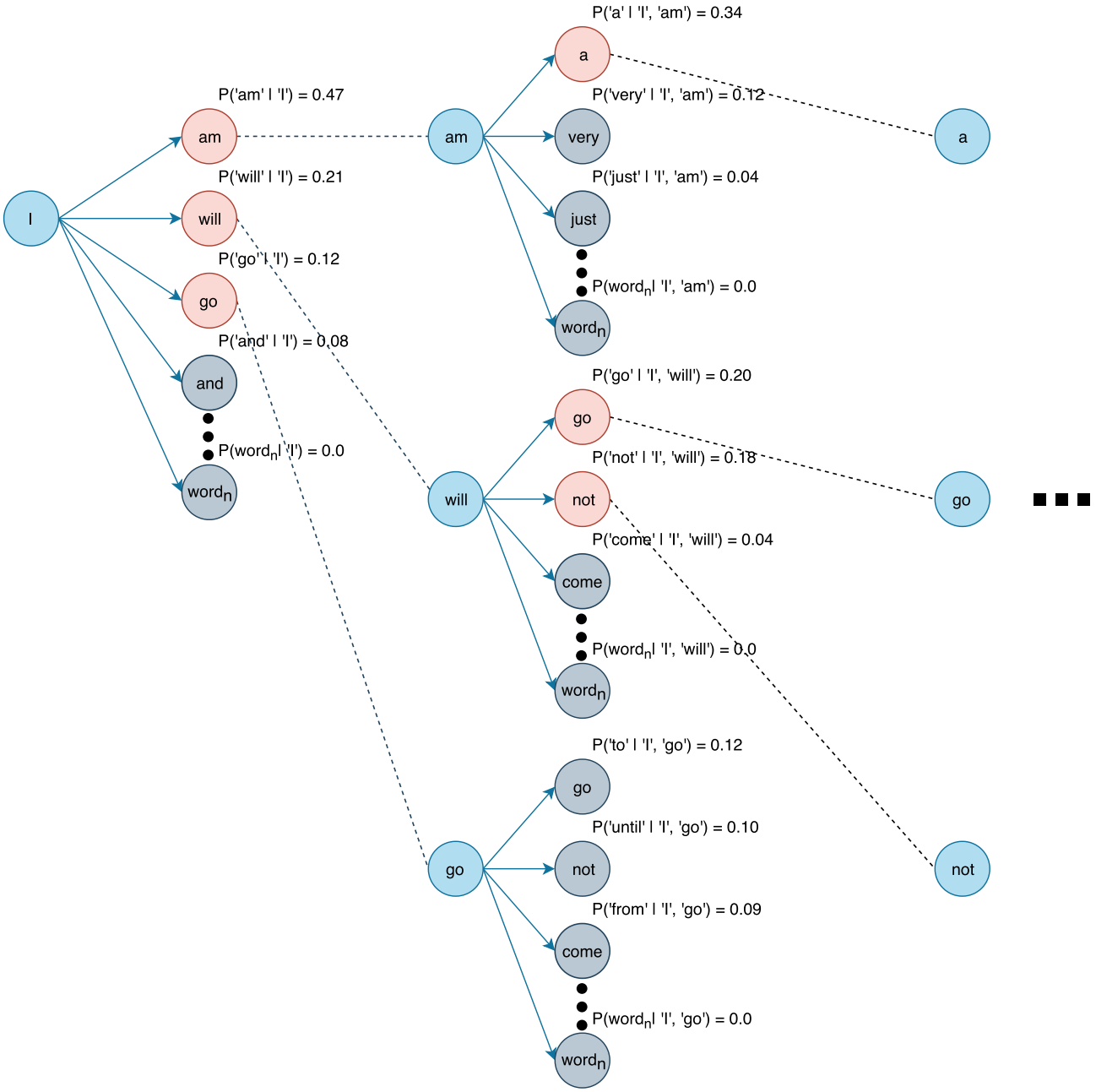


Fig. 7 A sample diagram for beam search while choosing next word with beam width 3. Figure best viewed in color

decaying learning rate is used for the model, starting with a value of 10^{-3}. ADAM optimizer is used for applying the gradients according to the loss. The drop-out technique is not used for this model. The training lasted for 32 epochs. Two models are trained with the same configurations; one for left-leanings and one for the other side. For both models, training sets are an equal size of 750,000 tweets. Generated tweets are sent to the classifier explained in Sect. 5. The tweets with the highest probabilities are used from both real and generated tweets during the evaluation.

As a baseline, two different models are prepared. We generate tweets randomly and with maximum likelihood estimation (MLE) model. For a random generation, all tweets are represented with part of speech tag sequences. Vocabulary for each tag is a set of all words observed for that tag. Forming a sentence starts with selecting a random POS structure. A random word from the vocabulary for the POS tag is assigned. MLE model is trained with the tweets in order to extract the probability distribution with trigrams of words.

Human evaluators evaluate generated tweets in two criteria. Namely, we want from our participants to score the tweets according to *Meaning* and *Quality* between the range of (1–5) where 5 refers the best. *Meaning* score denotes whether the tweet reflects the view of the discussion side. *Quality* of the tweet is whether it looks like a human-posted tweet. Five human evaluators assess each tweet, and 32 tweets are used for each model in the experiment. In order to compare with the original tweets, 32 original tweets are also scored by them.

Table 6 Human evaluation results for generated and real tweets

Side	Source	Meaning	Quality
Right-leaning	Randomly generated	1.69	1.91
	MLE	2.81	2.97
	Proposed model	3.55	3.56
	Real accounts	3.73	3.73
Left-leaning	Randomly generated	1.69	1.88
	MLE	2.53	3.09
	Proposed model	3.62	3.52
	Real accounts	3.66	3.77

Table 6 shows that the proposed model has the highest meaning and quality scores among the generated tweets. Moreover, generated tweets of our model are scored close to the human-posted tweets. The *Quality* score of the tweet is fundamental for the bot to deceive other users easily. Our model also has comparable *Meaning* score relative to the real ones. It means that the model can produce tweets that reflect the bot's side in the debate. Since we aim to develop a bot that supports a side in this study, together with *Quality*, *Meaning* occupies a vital place.

Additionally, we prepared a multiple-choice test where containing three real user tweets and one generated tweet, as shown in Table 7 to check the indistinguishability of our tweets. The model can generate tweets that are highly similar to the real ones. The evaluator is expected to pick out the generated tweet. There are 30 questions where 7 participants answer each one.

Table 8 shows that our model is more successful than random and maximum likelihood models. Evaluators are not able to detect the generated tweet in an average of 60.95% and 65.71% times for right-leaning and left-leaning tweets. However, this statistic gives us evaluator-oriented performance. We examined each question to see the generated

Table 7 Sample questions on multiple-choice test for finding the generated tweet from among the real ones

Choices	Leaning
(a) Im so tired of this drama king ! ⟨url⟩	
(b) im so tired of the left s face of bigotry and hate of the nra. ⟨hashtag⟩	
(c) Im so tired of hearing the term assault weapons	
(d) Im so tired of hearing about ⟨hashtag⟩ it didnt work today and it wont work tomorrow	Right
(a) This is why people need the right to protect themselves from the liberal left. ⟨url⟩	
(b) This is why people need to arm themselves. period ! ⟨url⟩	
(c) This is why people need to prosecuted for ⟨hashtag⟩ ⟨hashtag⟩	
(d) This is why people need to be heard ! ⟨url⟩	Right
(a) Unlike the ⟨hashtag⟩ started pointing out what to form the constitution.	
(b) Unlike the ⟨hashtag⟩ crowd who digs in and gets it done on behalf of everyone s ⟨hashtag⟩ ⟨url⟩	
(c) Unlike the ⟨hashtag⟩ these heroic cops ran toward the florida school shooting. ⟨hashtag⟩ ⟨hashtag⟩ ⟨url⟩	
(d) Unlike the ⟨hashtag⟩ hollywood does promote criminal ⟨hashtag⟩ and ⟨hashtag⟩	Right
(a) ⟨user⟩ you mean like the cop that was at ⟨hashtag⟩	
(b) ⟨user⟩ you mean like those poor kids that were slaughtered in ⟨hashtag⟩. ⟨url⟩	
(c) ⟨user⟩ you mean that you are not a gun owner. ⟨hashtag⟩ ⟨hashtag⟩	
(d) ⟨user⟩ you mean warriors of the left ? ⟨hashtag⟩ ⟨hashtag⟩	Left
(a) This is about a hard decision. god bless our kids and teachers ⟨hashtag⟩ ⟨hashtag⟩	
(b) This is about guns. guns. if you cant see it you dont want to see it. ⟨hashtag⟩ ⟨hashtag⟩ ⟨url⟩	
(c) This is about ⟨hashtag⟩ for our ⟨hashtag⟩ reveal plan to destroy ⟨hashtag⟩ in bed with ⟨hashtag⟩	
(d) This is about capturing a ground swell of emotion against the nra and for stricter gun control laws ⟨user⟩ on ⟨hashtag⟩ ⟨url⟩	Left
(a) Children are our future has never been more true. humbled and grateful for the ⟨hashtag⟩ students speaking truth to power ⟨ hashtag⟩ ⟨url⟩	
(b) children are our future! you stand the ⟨hashtag⟩ and stop taking money from the nra. do the right thing or ⟨hashtag⟩	
(c) Children are our future! please support them!!! ⟨hashtag⟩ ⟨hashtag⟩ ⟨url⟩	
(d) Children are our future. but how can they have a future if our gun laws dont change ? ⟨hashtag⟩ ⟨hashtag⟩ ⟨url⟩	Left

Generated tweets are marked in bold

Table 8 Average of non-disclosed tweets for each human evaluator result on multiple-choice test for finding the generated tweet among the real ones

	Right-leaning Non-disclosed	Left-leaning Non-disclosed
Random	13.33 %	16.67 %
MLE	33.33 %	48.00 %
Proposed model	**60.95 %**	**65.71 %**

The best results are shown in bold

tweets' success. In Table 9, the ratio of the tweets that participants cannot find the generated one on multiple-choice tests is listed. The correct participants column shows that if these many evaluators find the generated tweet for a question, then we count it as revealed. None of the tweets generated by our model is revealed by more than 75% of the evaluators at the same time.

7 Reply generation to an opposite view

To mimic real Twitter accounts, posting meaningful tweets is very important, but not enough. It should also interact with other users to support its idea. Interaction in Twitter is done with two mechanisms: reply and mention. Mention is used if a user wants a specific user to get notification indicates that she/he is talking about something related. Tagging real user accounts just before or after the generated tweet is adequate. Reply task is unlike the tweet generation which is covered in the previous section. The given answer should be related to the origin post.

Starting text generation before seeing the whole text comes up with unrelated output tweets. The model needs to be fed by the origin tweet until the end to catch the correct meaning. Therefore, we pick the encoder–decoder model. The encoder part is in charge of extracting the meaning from the source tweet. The relation between the arrangement of words and the semantic is encoded into latent variables. The decoder part is subject to select the next words for forming a sentence based on the latent variables.

This model is very popular for addressing several NLP problems including text generation such as question answering and neural machine translation (Wang and Nyberg 2015; Cho et al. 2014). It can comprehend the given sequence and produce a new one from it.

There are some metrics such as BLEU (Papineni et al. 2002) and METEOR (Denkowski and Lavie 2011) for measuring the quality of machine translation. They are also used for dialogue quality widely (Sordoni et al. 2015; Huber et al. 2018; Luo et al. 2018). However, they are word-overlap metrics and restrict creativity. These metrics are more suitable for translation quality (Adiwardana et al. 2020). We propose a new metric kn-BLEU, which is based on BLEU. BLEU measures the quality of the translation by comparing a candidate translation to the reference translations. It gets trendy because of being fast, inexpensive, and language-independent (Papineni et al. 2002). Since there may be alternatives to correct translation, it can get multiple reference translations to compare. The score is based on searching the n-grams of the candidate for reference translations and their length. The calculation ignores the word order. Unigrams can be used in this metric. Some studies include human evaluation results accordingly (Li et al. 2016; Wen et al. 2017; Hashimoto et al. 2019). Human evaluators and kn-BLEU are used for measuring the performance of our model.

7.1 Extracting the meaning from the Tweets

The dataset consists of source tweets as inputs and replies as labels; $D = (S, R)$, where $S = \langle s_1, s_2, s_3, \ldots, s_N \rangle$ and $R = \langle r_1, r_2, r_3, \ldots, r_N \rangle$. Words in the sentences are represented with GloVe embeddings just we did in other sections. We use the same notation for representations of the tweets here as $\langle v_{s1}, v_{s2}, v_{s3}, \ldots, v_{sm} \rangle$ for source and $\langle v_{r1}, v_{r2}, v_{r3}, \ldots, v_{rm} \rangle$ for reply.

The encoder gets a source tweet as one word at a time. At each time, the model is fed with v_{si}, and the state is updated accordingly. While getting new embeddings as input, the meaning starts to be encoded into the state vector,

Table 9 Non-disclosed tweets from human evaluation results on multiple-choice test for finding the generated tweet among the real ones

Correct participants (%)	Right-leaning			Left-leaning		
	Random (%)	MLE (%)	Proposed model (%)	Random (%)	MLE (%)	Proposed model (%)
25	0.0	0.0	**26.7**	0.0	**26.7**	20.0
50	0.0	26.7	**73.3**	0.0	53.3	**66.7**
75	13.3	60.0	**100.0**	13.3	33.3	**100.0**
100	40.0	86.7	**100.0**	53.3	86.7	**100.0**

The best results are shown in bold

$h_i = f(h_{i-1}, v_{si})$ where f is the abstract function representation of the network. Unlike the model we applied in Sect. 6, we do not work on intermediate states because the meaning is not concluded yet. Since one word can directly turn the meaning to the opposite in natural languages, the model should wait until the sentence is over. The resulting state represents the meaning of the tweet and can be used for various purposes. In Sect. 5, we used this encoding intending to classify the sentiment. In this section, we use it to generate replies to it.

We use another sequence of cells, called decoder, for text generation. It gets the last state of the LSTM cells as input and forms a sentence that is a reply to it. In addition to the encoder's final state, we applied the attention mechanism, retrieved from Luong et al. (2015), to our model for getting more related replies. Instead of using only the encoding, the model minds the specific input vectors more based on the learned attention weights. This information helps the model to generate the replies within the context. Figure 8 visualizes the the model architecture in details with an example.

7.2 Generating reply according to the encoding

Extracted features are fed into the second part of our model, decoder. It is responsible for forming a new sentence conforming with the source tweet. The procedure is done basically with predicting one word at a time. The following word depends on the encoder state, attention vector, and the current state of the decoder.

Similar to Sect. 6, the prediction of the following word is made among the vocabulary. The model assigns a probability of occurrence for each word in the vocabulary and selects the most probable one. Hence, we apply the softmax activation function to get probability values from logits. Sequence loss is used for optimizing our model. It sums up the cross-entropy loss across the predicted sequence. The weights of the loss along the sequence are set equal since the meaning depends on all words equally.

We applied dropout to our model to make it more robust to over-fitting. Furthermore, we used it to add randomness to our model while sampling replies. Since LSTM networks are very powerful, it is prone to generate memorized sentences. It makes our bot weak in hiding itself. We observed that using dropout on the generation phase is also useful. It changes the followed path from an ordinary one to the original but still related one.

7.3 kn-BLEU: new metric on measuring the relatedness

For training a Twitter bot that can participate in discussions, the BLEU score does not fit perfectly to measure the quality of the reply. This inconsistency is mainly because of that we cannot talk about a reference replies for the tweets. Besides, we want our bot to produce original tweets that spoil the metric. Therefore, we propose a new metric kn-BLEU where kn stands for k-nearest. It is based on BLEU. Instead of comparing the generated translations with targeted ones, we widen the space of references with similar content. Since we want to measure the relatedness of the generated tweet, it is valid for our purpose. Chosen words can give a thought about the

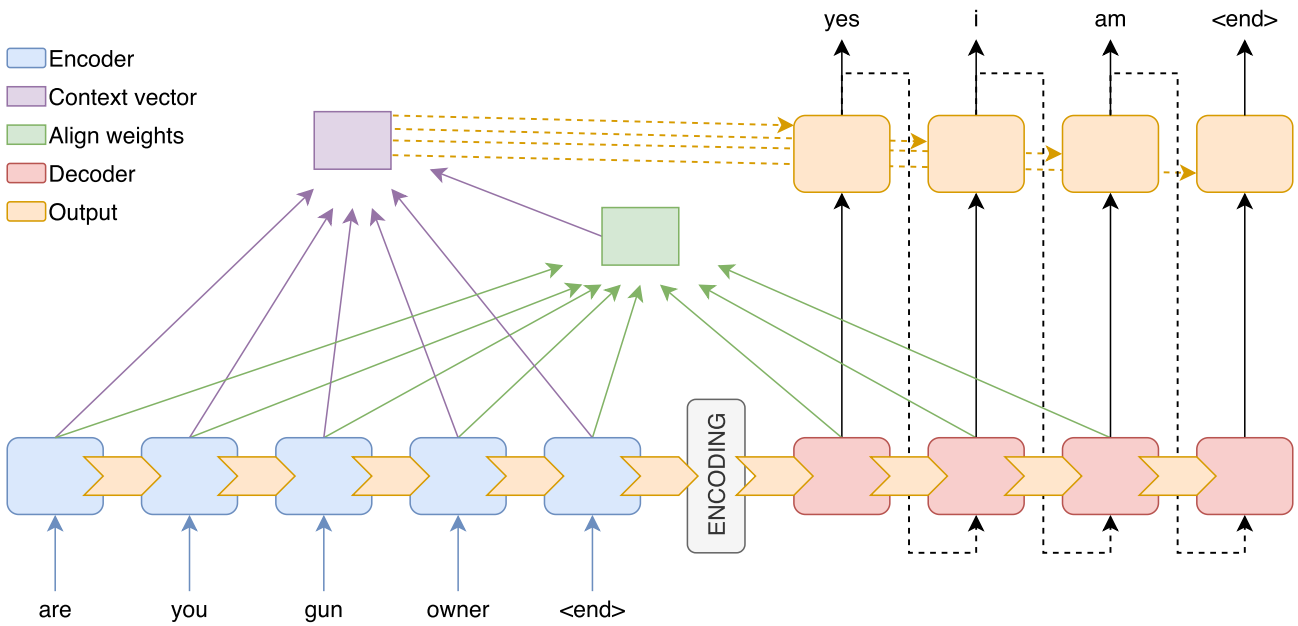


Fig. 8 Encoder–decoder architecture with attention mechanism for reply generation. Figure best viewed in color

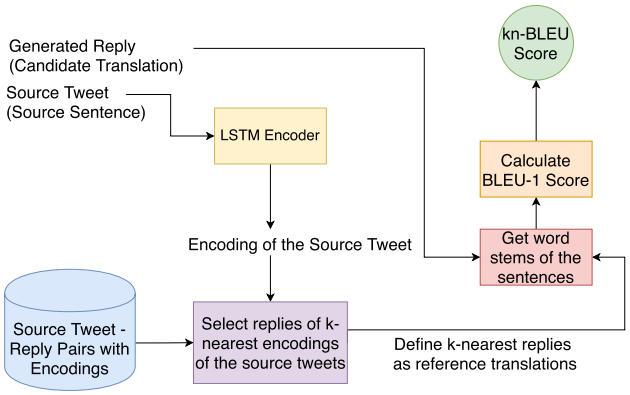


Fig. 9 Flow of kn-BLEU metric that measures the relatedness of the translation. Generated reply is equivalent of the candidate translation in neural machine translation domain and the retrieved replies are the reference translations

context. Hence, we worked on unigrams while testing our metric. Figure 9 visualizes the flow of proposed metric.

The main procedure is encoding the sentences into fixed-length vectors. We used the encoder–decoder LSTM model for this aim and preferred GloVe for representing tweets. The model is trained for regenerating the text itself. It moves the sentences from natural languages to the hypothetical context space. We used this space to find the sentences with the closest semantic to the source sentence. To get the kn-BLEU score of a generated reply, we first retrieve the closest context vectors of the source sentence from our database that is used for training and before. The replies given to these source sentences are selected as reference translations called in the BLEU context. The words are stemmed in both generated reply and reference translations. BLEU-1 score of the generated reply is calculated and emitted.

To establish the functionality of the approach, we tested the metric in the neural machine translation task. The encoder is trained with the source sentences of the WMT news-test-2014 dataset in the German language. We observed that the closest sentences might have similar meanings or words. English translations of the closest five sentences of some examples are listed in Table 10. The model can encode the context into the vector successfully.

Since we want to measure the word selection quality of our model, unigrams are compared. Moreover, the same word can exist in several different forms. We are not interested in the form of the word. Thus, we stemmed the words before applying BLEU on the sentences. The following examples are retrieved from neural machine translation (seq2seq) tutorial to visualize the need for stemming (Luong et al. 2017). For one of the sentences, reference translation

Table 10 k-nearest context to the given sentence with the WMT news-test-2014 test dataset where k is 5

Source sentence	Closest sentences
Maybe we ’re more “Mexican” than I thought	(1) We ’re not FC Barcelona! (2) This year, we ’re really committed to winning the championship. (3) In Mexico, we ’re seen as Americans. (4) “We get treated like criminals, ”Mr. Orozco said. (5) However, Mexican law recognises the possibility.
“ Weapons were stolen by Russian officers and by the Caucasians, ” says Baranets	(1) There were twenty of us, including Sharon Stone and John Kennedy Jr. (2) Several bills were blocked by vetoes of Democratic governors. (3) Next are “ black weapons, ” stolen by criminals from representatives of defence agencies. (4) Both projects were destined for military use of rocket engines. (5) Manufacturers of weapons make their contribution, according to Baranets
The Ministry of the Interior does not put arms from the illegal market back into circulation	(1) Nonetheless , the Ministry of the Interior asserts that the situation of the spread of illegal arms is under control. (2) The eight planets of our solar system, plus the dwarf planet Ceres (3) This image was put together from the X-ray images captured by various telescopes (4) The Russian Ministry of the Interior is proposing to toughen up the law for owners of civil weapons (5) The latter caused a sensation by explaining the mechanics of the public contracts “system ” to the commission
- What do you mean ?	(1) Do you want him ?’ (2) How do you select them ? (3) What now ? (4) - What do you think about voice control ? (5) How do you explain this progression ?
It ’s not an easy task	(1) And that ’s not all (2) That ’s not going to happen (3) There ’s an awful lot of questions (4) That ’s fair (5) That ’s a difficult one

is *"Republican leaders justified their policy by the need to combat electoral fraud"* and the neural machine translation is *"Republicans are justifying their policy with the need to combat electoral fraud"*. The machine-translated version can be accepted as perfect in the context manner. However, the BLEU score is low because of the difference in the forms of the words. BLEU-1 score with the original translation is 69.23, but the stemmed BLEU-1 score is 84.62.

We applied both BLEU-1 and kn-BLEU with $k = 5$ to the machine translation outputs of the pre-trained model in Luong et al. (2017). To compare them, we applied stemming before getting the BLEU-1 score. Our proposed method gives an average score of 64.72, where the BLEU-1 is 53.65. Increased three and stayed the same three samples are given in Table 11. The results are promising in measuring the context relevancy. The new metric is not a suitable tool for grammatically comparing translation quality. However, it gives consequential information about the relatedness to the context over selected words.

7.4 Flow of replying opposite view Tweets

There is an automated flow for searching dissident tweets on Twitter and post a reply to them. This flow focuses on generating highly detailed replies to the target tweet. Also, produced tweets should be noncontradictory with their point of view. Namely, if the bot is supposed to be left-leaning, then it should not generate pro-gun content. Our bot collects tweets across Twitter by given tokens. The mentions directed to our bots are also included. It does not interact with all of these tweets.

First, it filters them according to the pre-trained classifier that is explained in Sect. 5. If the classifier predicts the label of the given tweet with the probability higher than the given threshold, then it is taken into account. Other tweets are accepted as the bot is not sure and ignored.

After choosing which tweets should be replied, we feed our model with these tweets to generate replies. The model emits new sequences of words to form tweets, and produced tweets need to be checked before getting posted. Our bot checks the generated tweets under the desired quality in terms of both relatedness and the meaning manners. In other words, if the reply is not consistent with the target tweet, it does not post it. The relatedness check is done with kn-BLEU metric, which is described in Sect. 7.3. It gets the tweets to reply and produce a reply to each of them. Generated replies are checked, and the ones that passed the relatedness check are returned.

7.5 Experiments

Similar to other models, we used GloVe embeddings, which are 200-dimensional vectors, and trained with the Twitter dataset to symbolize words. Missing words in the GloVe are tagged with special keyword $\langle unk \rangle$. The architecture

Table 11 BLEU-1 and kn-BLEU scores on increased 3 and stayed the same 3 samples on news-test-2014 test dataset where k is 5

Reference translation	Machine translation	BLEU-1	kn-BLEU
Manning testified Thursday about his arrest in Iraq and his transfer to Kuwait, where he was held for nearly two months before being transferred to the brig at Marine Base Quantico in Virginia in July 2010	On Thursday, his arrest and postponement to Kuwait, where he held almost two months before he was imprisoned in July 2010, was held in the prison of the navy of Virginia	53.46	78.13
It was through him that the scandal broke in 2011, in an in-depth investigation into corruption related to road construction contracts in Quebec, to which the liberal Prime Minister at the time, Jean Charest, had consented only reluctantly	The scandal was revealed by an in-depth investigation into the untrustworthiness of the contracts on road construction in Quebec, which the then prime minister had reluctantly agreed to	51.29	75.86
A referendum is due to be held within the next 2 weeks	A referendum should be held in 14 days	35.62	66.67
In accordance with the wishes of the military, the Defence budget review will be not submitted to Parliament, but to a National Defence Council	In line with the military' s wish, the review of the defence budget will not be presented to Parliament, but to a national defence council	85.19	85.19
The shelves are full to capacity	The shelves are filled up to the top	55.56	55.56
Increasingly, an unrelieved patient will have the option of having such palliative sedation	Patients without pain reduction will have an increasingly common ability to back this date	0.4	0.4

Table 12 Average kn-BLEU scores of generated replies for both sides of the debate with $k = 5$

Side	Source	*kn*-BLEU
Right-leaning	Real accounts	43.72
	Proposed model	44.98
Left-leaning	Real accounts	44.51
	Proposed model	41.64

Table 13 Human evaluation results for generated and real replies

Side	Source	Relatedness	Quality
Right-leaning	Proposed model	4.07	3.62
	Real accounts	4.24	3.76
Left-leaning	Proposed model	4.14	3.56
	Real accounts	3.92	3.43

we applied is one layer LSTM with a size of 256. It consists of two parts, namely encoder and decoder. Dropout is applied to the cells to prevent memorize the data and adding randomness while sampling. ADAM optimizer with 10^{-2} initial learning rate is used for training. Two models are trained separately for two views of the debate, similar to Sect. 6. Configurations for these models are identical. Learning lasted for 150 epochs. For filtering, the produced tweets meaning threshold is fixed to 85%, and the kn-BLEU threshold is set to 45. In addition to training, we used dropout for sampling. The dropout rate for sampling is 50%.

The training set and test set contain 250,000 and 65,000 tweets and reply tuple, respectively, for each model. Produced results showed that the LSTM model could over-fit on training data and produce the same replies again. For measuring kn-BLEU of generated replies, we train another model to produce the tweet itself with these 250,000 training set tweets. k of the kn-BLEU is set to 5 for the experiments. 3000 different tweets are produced for both sides. Before filtering unrelated replies out, we get the average kn-BLEU scores for both models. Table 12 shows these results. If the score of a reply is 100, then it is the same with one of k-nearest tweets in the training set. It is 0 if any of the words are not included in any of them. The kn-BLEU of generated replies, 44.98 for right-leaning replies, and 41.64 for left-leaning replies show that the model can produce highly detailed replies to the target tweets.

Human evaluation is done on generated and real replies like we did in Sect. 6. 60 tweets are randomly chosen from the tweets passed the threshold. Real and generated right-leaning and left-leaning replies are with equal proportion. *Relatedness* and *Quality* are two scores that we want participants to mark. Scores are in scale between 1 and 5, where 5 stands for the best. Table 13 includes the results of human evaluation. *Relatedness* refers to how the given reply is related to the target tweet. *Quality* is the measure if the generated reply looks like a human-generated one. The scores of the proposed model for left-leaning tweets are slightly higher than real accounts. Quality scores are similar to the generated tweet evaluation experiments described in Sect. 7. Relatedness score is promising for both models, which implies that our model can capture the meaning and generate a reply related to it.

8 Conclusions

In this study, we propose deep learning models for developing a chatbot that can meaningfully participate in a debate on Twitter by taking a side. The chatbot is capable of detecting a tweet's position in a debate, generating new topic-related supportive tweets, and replying to a tweet from the opposing side.

The first model that is developed is a tweet classifier that returns the probability of which side in a debate (i.e., pro vs. anti) it belongs to. The evaluations presented in Tables 3 and 4 show that the model is capable of predicting the label correctly most of the time. Furthermore, narrowing the tweets through their predicted probability as a threshold increases the accuracy of the approach.

The tweet generation model uses LSTM as its main component. Generated tweets mostly reflect the political position of the chatbot. However, sometimes it gets confused. In order to minimize this deficiency, the classifier model is used for scoring and filtering the generated tweets. If a generated tweet does not match its intended position correctly, then it is discarded.

The third model that is developed for reply generation includes an encoder–decoder LSTM architecture. Its input is a tweet from the other side that needs a reply. Like the tweet generation model, sometimes, the reply generator fails to generate a related reply or carry the correct bias. The classifier model is again applied to the generated reply to assess it for the intended bias. Relatedness issue is handled via a new metric, named kn-BLEU. The generated tweet is discarded if it fails to meet the experimentally determined accuracy thresholds on either of these measures.

The accounts in our dataset generally tweet about the US gun debate. A chatbot that can tweet about more diverse topics may lower its accuracy and coherence. Generated tweets contain some special tags, such as ⟨*number*⟩, ⟨*hashtag*⟩, and ⟨*user*⟩. As future work, filling these placeholders with a relevant entity is crucial. Even though we use human evaluation to determine whether our chatbot can be easily spotted as anomalous or not, the best way to test it would be online and by evaluating if it can achieve high levels of engagement (i.e., likes, shares, follows) and important effects,

such as consensus, moderation, and tolerance from its target audiences.

References

Adiwardana D, Luong M-T, So DR, Hall J, Fiedel N, Thoppilan R, Yang Z, Kulshreshtha A, Nemade G, Lu Y (2020) Towards a human-like open-domain chatbot. arXiv preprint arXiv :2001.09977

Antipov G, Baccouche M, Dugelay J (2017) Face aging with conditional generative adversarial networks. In: 2017 IEEE international conference on image processing, ICIP 2017

Bahdanau D, Cho K, Bengio Y (2015) Neural machine translation by jointly learning to align and translate. In: 3rd international conference on learning representations, ICLR 2015, conference track proceedings

Blei DM, Ng AY, Jordan MI (2003) Latent dirichlet allocation. J Mach Learn Res 3:993–1022

Bojanowski P, Grave E, Joulin A, Mikolov T (2017) Enriching word vectors with subword information. Trans Assoc Comput Linguist 5:135–146

Bradeško L, Mladenić D (2012) A survey of Chatbot systems through a Loebner prize competition. In: Proceedings of Slovenian language technologies society eighth conference of language technologies, pp 34–37

Brandtzæg PB, Følstad A (2018) Chatbots: changing user needs and motivations. Interactions 25(5):38–43

Brixey J, Hoegen R, Lan W, Rusow J, Singla K, Yin X, Artstein R, Leuski A (2017) Shihbot: a facebook chatbot for sexual health information on HIV/AIDS. In: Jokinen K, Stede M, DeVault D, Louis A (eds) Proceedings of the 18th annual SIGdial meeting on discourse and dialogue, Saarbrücken, Germany, August 15–17, 2017, Association for Computational Linguistics, pp 370–373

Chavoshi N, Hamooni H, Mueen A (2016) Identifying correlated bots in twitter. In: Social informatics—8th international conference, proceedings, Part II, volume of 10047 of lecture notes in computer science, pp 14–21

Chen H, Liu X, Yin D, Tang J (2017) A survey on dialogue systems: recent advances and new frontiers. SIGKDD Explor 19(2):25–35

Chen C, Mu S, Xiao W, Ye Z, Wu L, Ju Q (2019) Improving image captioning with conditional generative adversarial nets. In: The thirty-third AAAI conference on artificial intelligence, AAAI 2019, the thirty-first innovative applications of artificial intelligence conference, IAAI 2019, the ninth AAAI symposium on educational advances in artificial intelligence, EAAI 2019, AAAI Press, pp 8142–8150

Chollampatt S, Ng HT (2018) A multilayer convolutional encoder-decoder neural network for grammatical error correction. In: Thirty-second AAAI conference on artificial intelligence

Cho K, van Merrienboer B, Bahdanau D, Bengio Y (2014) On the properties of neural machine translation: Encoder-decoder approaches. In: Proceedings of SSST@EMNLP 2014, eighth workshop on syntax, semantics and structure in statistical translation, Association for computational linguistics, pp 103–111

Cui R, Agrawal G, Ramnath R (2020) Tweets can tell: activity recognition using hybrid gated recurrent neural networks. Soc Netw Anal Min 10(1):1–15

Cui L, Huang S, Wei F, Tan C, Duan C, Zhou M (2017) Superagent: a customer service chatbot for e-commerce websites. In: Proceedings of the 55th annual meeting of the association for computational linguistics, ACL 2017, Association for Computational Linguistics, pp 97–102

Demirel B, Cinbis RG, Ikizler-Cinbis N (2019) Image captioning with unseen objects. In: 30th British machine vision conference

Denkowski MJ, Lavie A (2011) Meteor 1.3: automatic metric for reliable optimization and evaluation of machine translation systems. In: Proceedings of the sixth workshop on statistical machine translation, WMT@EMNLP 2011, Association for computational linguistics, pp 85–91

Dumais ST, Furnas GW, Landauer TK, Deerwester S, Harshman R (1988) Using latent semantic analysis to improve access to textual information. In: Proceedings of the SIGCHI conference on Human factors in computing systems, pp 281–285

Ebrahimi M (2016) Automatic identification of online predators in chat logs by anomaly detection and deep learning. Ph.D. thesis, Concordia University,

Ertugrul AM, Karagoz P (2018) Movie genre classification from plot summaries using bidirectional LSTM. In: 12th IEEE international conference on semantic computing, ICSC 2018, IEEE Computer Society, pp 248–251

Ertugrul AM, Velioglu B, Karagoz P (2017) Word embedding based event detection on social media. In: Hybrid artificial intelligent systems—12th international proceedings and conference, HAIS 2017, volume 10334 of lecture notes in computer science, Springer, New York, pp 3–14

Garimella K, Morales G, Gionis A, Mathioudakis M (2018) Political discourse on social media: Echo chambers, gatekeepers, and the price of bipartisanship. In: Proceedings of the 2018 World Wide Web Conference, pp 913–922

Goodfellow I (2016) Nips 2016 tutorial: Generative adversarial networks. arXiv preprint arXiv:1701.00160,

Goswami A, Kumar A (2016) A survey of event detection techniques in online social networks. Soc. Netw. Anal. Min. 6(1):107:1–107:25

Hashimoto TB, Zhang H, Liang P (2019) Unifying human and statistical evaluation for natural language generation. In: Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, NAACL-HLT 2019, Volume 1, Association for Computational Linguistics, pp 1689–1701

Higashinaka R, Imamura K, Meguro T, Miyazaki C, Kobayashi N, Sugiyama H, Hirano T, Makino T, Matsuo Y (2014) Towards an open-domain conversational system fully based on natural language processing. In: Proceedings of COLING 2014, the 25th international conference on computational linguistics: technical papers, pp 928–939

Hochreiter S, Schmidhuber J (1997) Long short-term memory. Neural Comput 9(8):1735–1780

Holotescu C (2016) Moocbuddy: a chatbot for personalized learning with MOOCS. In: Iftene A, Vanderdonckt J (eds) 13th International conference on human computer interaction, RoCHI 2016, Iasi, Romania, September 8-9, 2016, Matrix Rom, pp 91–94

Howard PN, Kollanyi B, Woolley SC (2016) Bots and automation over twitter during the second U.S. presidential debate. In: Data Memo 2016.2

Huang M, Zhu X, Gao J (2020) Challenges in building intelligent open-domain dialog systems. ACM Trans Inf Syst (TOIS) 38(3):1–32

Huber B, McDuff DJ, Brockett C, Galley M, Dolan B (2018) Emotional dialogue generation using image-grounded language models. In: Proceedings of the 2018 CHI conference on human factors in computing systems, CHI 2018, ACM, p 277

Hussain S, Sianaki OA, Ababneh N (2019) A survey on conversational agents/chatbots classification and design techniques. In: Web, artificial intelligence and network applications—proceedings of the workshops of the 33rd international conference on advanced information networking and applications, AINA workshops 2019, vol 927 of advances in intelligent systems and computing, Springer, New York, pp 946–956

Karim F, Majumdar S, Darabi H, Chen S (2018) LSTM fully convolutional networks for time series classification. IEEE Access 6:1662–1669

Karras T, Laine S, Aila T (2019) A style-based generator architecture for generative adversarial networks. In: IEEE conference on computer vision and pattern recognition, CVPR 2019, Computer vision foundation/IEEE, pp 4401–4410

Kim J, Oh S, Kwon O-W, Kim H (2019) Multi-turn chatbot based on query-context attentions and dual wasserstein generative adversarial networks. Appl Sci 9(18):3908

Lei W, Jin X, Kan M, Ren Z, He X, Yin D (2018) Sequicity: simplifying task-oriented dialogue systems with single sequence-to-sequence architectures. In: Proceedings of the 56th annual meeting of the association for computational linguistics, ACL 2018, Volume 1, Association for computational linguistics, pp 1437–1447

Li J, Galley M, Brockett C, Spithourakis GP, Gao J, Dolan WB (2016) A persona-based neural conversation model. In: Proceedings of the 54th annual meeting of the association for computational linguistics, ACL 2016, Volume 1, The Association for Computer Linguistics

Li J, Monroe W, Ritter A, Jurafsky D, Galley M, Gao J (2016) Deep reinforcement learning for dialogue generation. In: Proceedings of the 2016 conference on empirical methods in natural language processing, EMNLP 2016, The Association for Computational Linguistics, pp 1192–1202

Li J, Monroe W, Shi T, Jean S, Ritter A, Jurafsky D (2017) Adversarial learning for neural dialogue generation. In: Palmer M, Hwa R, Riedel S (eds) Proceedings of the 2017 conference on empirical methods in natural language processing, EMNLP 2017, Copenhagen, Denmark, September 9–11, 2017, Association for Computational Linguistics, pp 2157–2169

Liu ILB, Cheung CMK, Lee MKO (2010) Understanding twitter usage: What drive people continue to tweet. In: Pacific Asia conference on information systems, PACIS 2010, AISeL, p 92

Luong M, Brevdo E, Zhao R (2017) Neural machine translation (seq2seq) tutorial. https://github.com/tensorflow/nmt

Luong T, Pham H, Manning CD (2015) Effective approaches to attention-based neural machine translation. In: Proceedings of the 2015 conference on empirical methods in natural language processing, EMNLP 2015, The association for computational linguistics, pp 1412–1421

Luo L, Xu J, Lin J, Zeng Q, Sun X (2018) An auto-encoder matching model for learning utterance-level semantic dependency in dialogue generation. In: Proceedings of the 2018 conference on empirical methods in natural language processing, Association for Computational Linguistics, pp 702–707

Mahapatra A, Srivastava N, Srivastava J (2012) Contextual anomaly detection in text data. Algorithms 5(4):469–489

Mikolov T, Grave E, Bojanowski P, Puhrsch C, Joulin A (2018) Advances in pre-training distributed word representations. In: Proceedings of the international conference on language resources and evaluation (LREC 2018)

Mikolov T, Sutskever I, Chen K, Corrado GS, Dean J (2013) Distributed representations of words and phrases and their compositionality. In: Advances in neural information processing systems 26: 27th annual conference on neural information processing systems 2013. pp 3111–3119

Mondal A, Dey M, Das D, Nagpal S, Garda K (2018) Chatbot: an automated conversation system for the educational domain. In: 2018 international joint symposium on artificial intelligence and natural language processing (iSAI-NLP), IEEE, pp 1–5

Mostaço GM, De Souza ÍRC, Campos LB, Cugnasca C E (2018) Agronomobot: a smart answering chatbot applied to agricultural sensor networks. In: 14th international conference on precision agriculture, vol 24, pp 1–13

Neff G, Nagy P (2016) Automation, algorithms, and politics| talking to bots: symbiotic agency and the case of tay. Int J Commun 10:17

Papineni K, Roukos S, Ward T, Zhu W (2002) Bleu: a method for automatic evaluation of machine translation. In: Proceedings of the 40th annual meeting of the association for computational linguistics, ACL, pp 311–318

Parthornratt T, Kitsawat D, Putthapipat P, Koronjaruwat P (2018) A smart home automation via Facebook chatbot and raspberry pi. In: 2018 2nd International conference on engineering innovation (ICEI), IEEE, pp 52–56

Peng Y, Jiang H (2016) Leverage financial news to predict stock price movements using word embeddings and deep neural networks. In: NAACL HLT 2016, The 2016 conference of the North American chapter of the association for computational linguistics: human language technologies, The association for computational linguistics, pp 374–379

Pennington J, Socher R, Manning CD (2014) Glove: global vectors for word representation. In: Proceedings of the 2014 conference on empirical methods in natural language processing, EMNLP 2014, a meeting of SIGDAT, a special interest group of the ACL, ACL, pp 1532–1543

Qiu M, Li F, Wang S, Gao X, Chen Y, Zhao W, Chen H, Huang J Chu W (2017) Alime chat: a sequence to sequence and rerank based chatbot engine. In: Proceedings of the 55th annual meeting of the association for computational linguistics, ACL 2017, Volume 2, Association for Computational Linguistics, pp 498–503

Quarteroni S, Manandhar S (2007) A chatbot-based interactive question answering system. In: Proceedings of the 11th workshop on the semantics and pragmatics of dialogue, Decalog, pp 83–90

Radford A, Wu J, Child R, Luan D, Amodei D, Sutskever I (2019) Language models are unsupervised multitask learners. OpenAI Blog 1(8):9

Radford A, Metz L, Chintala S (2016) Unsupervised representation learning with deep convolutional generative adversarial networks. In: 4th international conference on learning representations, ICLR 2016, conference track proceedings

Rakshit G, Bowden KK, Reed L, Misra A, Walker MA (2017) Debbie, the debate bot of the future. In: Advanced social interaction with agents—8th international workshop on spoken dialog systems, IWSDS 2017, volume 510 of lecture notes in electrical engineering, Springer, New York, pp 45–52

Ramesh K, Ravishankaran S, Joshi A, Chandrasekaran K (2017) A survey of design techniques for conversational agents. In: International conference on information, communication and computing technology, Springer, New York, pp 336–350

Roca S, Sancho J, García J, Iglesias ÁA (2020) Microservice chatbot architecture for chronic patient support. J Biomed Inf 102:103305

Rosenstiel T, Sonderman J, Loker K, Ivancin M, Kjarval N (2015) Twitter and the news: How people use the social network to learn about the world. American Press Institute, Reston

Ruder S (2016) An overview of gradient descent optimization algorithms. CoRR, vol. abs/1609.04747

Sarikaya R (2017) The technology behind personal digital assistants: an overview of the system architecture and key components. IEEE Signal Process Mag 34(1):67–81

Shang L, Lu Z, Li H (2015) Neural responding machine for short-text conversation. In: Proceedings of the 53rd annual meeting of the association for computational linguistics and the 7th international joint conference on natural language processing of the Asian Federation of natural language processing, ACL 2015, Volume 1, The association for computer linguistics, pp 1577–1586

Shum H-Y, He X-d, Li D (2018) From eliza to xiaoice: challenges and opportunities with social chatbots. Front Inf Technol Electron Eng 19(1):10–26

Sordoni A, Galley M, Auli M, Brockett C, Ji Y, Mitchell M, Nie J, Gao J, Dolan B (2015) A neural network approach to context-sensitive

generation of conversational responses. In: NAACL HLT 2015, The 2015 conference of the North American chapter of the association for computational linguistics: human language technologies, The association for computational linguistics, pp 196–205

Su M-H, Wu C-H, Huang K-Y, Hong Q-B, Wang H-M (2017) A chatbot using LSTM-based multi-layer embedding for elderly care. In: 2017 international conference on orange technologies (ICOT), IEEE, pp 70–74

Sutskever I, Vinyals O, Le QV (2014) Sequence to sequence learning with neural networks. In: Advances in neural information processing systems 27: annual conference on neural information processing systems, pp 3104–3112

Tammewar A, Pamecha M, Jain C, Nagvenkar A, Modi K (2018) Production ready chatbots: generate if not retrieve. In: The workshops of the the thirty-second AAAI conference on artificial intelligence, vol. WS-18 of AAAI workshops, AAAI Press, pp 739–745

Tao C, Wu W, Xu C, Hu W, Zhao D, Yan R (2019) Multi-representation fusion network for multi-turn response selection in retrieval-based chatbots. In: Proceedings of the twelfth ACM international conference on web search and data mining, WSDM 2019, ACM, pp 267–275

Varol O, Ferrara E, Davis CA, Menczer F, Flammini A (2017) Online human-bot interactions: detection, estimation, and characterization. In: Proceedings of the eleventh international conference on web and social media, AAAI Press, pp 280–289

Wang S, Jiang J (2017) Machine comprehension using match-LSTM and answer pointer. In: 5th International conference on learning representations, ICLR 2017, conference track proceedings

Wang D, Nyberg E (2015) A long short-term memory model for answer sentence selection in question answering. In: Proceedings of the 53rd annual meeting of the association for computational linguistics and the 7th international joint conference on natural language processing (Volume 2: Short Papers), pp 707–712

Wang J, Yu L, Zhang W, Gong Y, Xu Y, Wang B, Zhang P, Zhang D (2017) IRGAN: a minimax game for unifying generative and discriminative information retrieval models. In: Kando N, Sakai T, Joho H, Li H, de Vries AP, White RW (eds) Proceedings of the 40th international ACM SIGIR conference on research and development in information retrieval, Shinjuku, Tokyo, Japan, August 7–11, 2017, ACM, pp 515–524

Weizenbaum J (1966) ELIZA—a computer program for the study of natural language communication between man and machine. Communications of the ACM 9(1):36–45

Wen T, Gasic M, Mrksic N, Su P, Vandyke D, Young S J (2015) Semantically conditioned LSTM-based natural language generation for spoken dialogue systems. In: Proceedings of the 2015 conference on empirical methods in natural language processing, EMNLP 2015, The Association for Computational Linguistics, pp 1711–1721

Wen T, Miao Y, Blunsom P, Young SJ (2017) Latent intention dialogue models. In: Proceedings of the 34th international conference on machine learning, ICML 2017, volume 70 of proceedings of machine learning research, PMLR, pp 3732–3741

Wu Y, Li Z, Wu W, Zhou M (2018) Response selection with topic clues for retrieval-based chatbots. Neurocomputing 316:251–261

Wu Y, Wu W, Xing C, Zhou M, Li Z (2017) Sequential matching network: a new architecture for multi-turn response selection in retrieval-based chatbots. In: Proceedings of the 55th annual meeting of the association for computational linguistics, ACL 2017, Volume 1, Association for computational linguistics, pp 496–505

Xu A, Liu Z, Guo Y, Sinha V, Akkiraju R (2017) A new chatbot for customer service on social media. In: Proceedings of the 2017 CHI conference on human factors in computing systems, ACM, pp 3506–3510

Yan R (2018) chitty-chitty-chat bot: Deep learning for conversational AI. In: Proceedings of the twenty-seventh international joint conference on artificial intelligence, IJCAI 2018, ijcai.org, pp 5520–5526

Yang Z, Chen W, Wang F, Xu B (2018) Improving neural machine translation with conditional sequence generative adversarial nets. In: Proceedings of the 2018 conference of the North American chapter of the association for computational linguistics: human language technologies, NAACL-HLT 2018, Volume 1, Association for Computational Linguistics, pp 1346–1355

Yin Z, Chang K, Zhang R (2017) Deepprobe: Information directed sequence understanding and chatbot design via recurrent neural networks. In: Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining, ACM, pp 2131–2139