3244-2010-0020 - Project Final Report

Ai Bo, Wang Yuchen, Zhang Xiaoyu, Muhammad Khairul Azman, Lim Yu Rong, Samuel, Goh Shau Cher Shaun
National University of Singapore
{ e0376958, e0261908, e0261883, e0309879, e0310716, e0175574 } @ u.nus.edu

Abstract

Misleading information in news articles poses a significant challenge to the integrity of public media consumption. This project therefore aims to find efficient and accurate ways of classifying fake news based only on the body text of articles. Both Deep Learning (DL) and traditional Machine Learning (ML) models are trained and tested on two fake news datasets, to evaluate their performance in this binary classification problem. It is found that ML models perform comparably with DL models on both datasets, and that performance varies significantly depending on the dataset used. Overall, ML and DL models are a feasible solution for classifying fake news articles, but this is contingent on building a diverse and well-rounded dataset for better generalization.

Introduction

"Fake news" is a term commonly used to describe misleading, and often entirely-false, news articles [1]. Such articles are generally propagated through social media, often by bots [2]. As a global media quandary and phenomenon, it has had a devastating impact on news media in general, and crucially on politics and public opinion. Due to the infeasibility of manually scanning each online news article, an automated solution is needed to solve this problem. One possible solution involves the use of Machine Learning techniques to identify fake news, using common semantic and syntactic traits in the articles' title and body text [3].

This project therefore aims to determine the feasibility of utilizing Machine Learning models in determining an article's authenticity from its textual features, specifically body text. Six common models are selected to benchmark on two publicly available datasets.

This paper will review current topical literature in Section 2 (Related Work). Dataset features and model specifics are discussed in Section 3 (Methodology). Experimental results are examined in Section 4 (Evaluation). Analysis of results is conducted in Section 5 (Discussion).

Related Work

This section evaluates the current existing literature on fake news detection using Machine Learning approaches. Recent literature focuses on two approaches to obtaining data: News-based approaches, and social context-based approaches.

News-based approaches. Features used for classification are usually retrieved from textual and visual aspects. Textual aspects include writing styles, emotions, and feelings of the news institutions, and visual aspects include pictures and recordings. For textual representation, a common strategy involves the application of tensor factorization methods [4] and Deep Learning based models [5][6][7]. For example, Ma et al. [8] deploy Recurrent Neural Networks to learn the representations of social media posts with time series.

Social context-based approaches. Features usually include information extracted from user profiles, comments and feedback. Some methods in social context-based fake news detection include user-based relations, semi-supervised detection, and unsupervised detection [9].

This project will follow a news-based approach to its analysis. Specifically, it will use Machine Learning-based models to analyze textual features of news articles.

Methodology

This section is an overview of the general approach used to appraise the feasibility of detecting the authenticity of a news article based on its textual features.

The models are grouped into traditional Machine Learning models (abbreviated as 'ML models'), and Deep Learning models (abbreviated as 'DL models'). Both ML and DL approaches are employed for this task.

Initial training and testing was performed on the ISOT dataset. The FNN dataset was then selected to supplement the

ISOT dataset due to the extremely high accuracies of 95% and above achieved on the ISOT dataset. Finally, models were benchmarked on the datasets, and additional experiments were conducted for further investigation.

3.1 Datasets

Two datasets are used for model training and testing. The first dataset is the ISOT dataset [10], collected by the Information Security and Object Technology (ISOT) research lab. The second dataset is the FakeNewsNet (FNN) dataset [11], collected by researchers from the University of Qom, and publicly available in the IEEE Dataport. The main attributes of these two datasets are summarized in the following table.

Table 1: Main attributes of ISOT and FNN datasets

	ISOT dataset	FNN dataset
Positive Instances (Real)	21417	8767
Negative Instances (Fake)	23481	8557
Average Text Word Count	406	4665
Origin of News Articles	*Reuters*	*PolitiFact.com*
Medium of News Articles	News	News and Social Media

Both datasets mainly consist of political news articles, from a diverse range of locations. These news articles are then collected by a news aggregator (either *Reuters* or *PolitiFact.com*). It should be noted that the articles aggregated by *PolitiFact.com* includes a wider range of sources like social media (Facebook, etc).

The FNN dataset was used in experiments to supplement the ISOT dataset, due to the unexpectedly-high testing accuracies on the ISOT dataset. The FNN dataset was chosen due to the average length of its articles, which are significantly longer than the articles from the ISOT dataset. This would result in a more diverse range of article content to train the models on.

3.2 Literature on Datasets

An accuracy of 99.8% has been achieved on the ISOT dataset using capsule neural networks with different levels of n-grams for feature extraction [12]. Separately, an accuracy of 92% has been achieved using the Term Frequency-Inverse Document Frequency approach to feature extraction and a Support Vector Machine as the classifier [13]. An accuracy of 88% has been achieved on the FakeNewsNet (FNN) dataset using a rule-based model for information analysis on multiple social media platforms by authors [14]. These values are considered the upper bounds for our model performances.

3.3 Data Pre-Processing

Raw text extracted from news articles could potentially yield noisy data due to its lack of structure. The text articles must thus be pre-processed to make the data suitable for training.

The following steps are used to pre-process the data used to train the three ML models. The text in the body of the article is first converted to lowercase. The text in the articles are then tokenized and separated into individual words and punctuations. Punctuation and stop words (such as 'a', 'in', and '.') in the tokenized text are then removed. Python's Natural Language Toolkit's list of stopwords was utilised for this purpose.

The pre-processing of data for the three Deep Learning models was performed by the respective library tokenizers.

3.4 Feature Extraction

For ML models, Term Frequency and Inverse Document Frequency vectors were extracted from article body texts and used as features.

Term Frequency represents the frequency of a term appearing in the article and is empirically calculated by dividing the number of times a word appears in the document by the total length of the document. Experimental results show that sublinear Term Frequency yields a higher accuracy in the ML models used. Sublinear Term Frequency is calculated by adding 1 to the logarithm of the Term Frequency.

$$TF(x,\ y)\ =\ \frac{Occurences\ of\ x\ in\ article\ y}{Total\ number\ of\ words\ in\ y}$$

$$SublinearTF(x,y) =\ 1\ +\ log(TF(x,y))$$

The Inverse Document Frequency is a measure of the relative rarity of a term across the entire dataset. The calculation is done by taking the logarithm of the ratio of total number of articles to the number of articles with the word in question.

$$IDF(x)\ =\ log(\frac{Total\ number\ of\ articles}{Number\ of\ articles\ with\ word\ x})$$

Finally, both Term Frequency and Inverse Document Frequency are vectorized and used as features for training. To simplify the feature extraction process, the *TfidfVectorizer* class from Python's *sklearn* library was used.

3.5 Models

Three traditional Machine Learning models and three Deep Learning models were selected for training and testing. The six models are described in more detail below.

3.5.1 Traditional Machine Learning (ML) models

Logistic Regression (LR)

While LR is generally less inclined to overfit, it can happen when the dimensions are high. To counteract this, L1 regularisation is utilized, encouraging learning from sparse features.

k-Nearest-Neighbours (kNN)

kNN is a classical ML method for classification. It is a non-parametric method and classifies sample points by a distance metric.

Support Vector Machine (SVM)

SVM was chosen due to its compatibility with NLP. SVM is highly effective in cases where the number of dimensions is greater than the number of samples which is usually the case in text classification.

3.5.2 Deep Learning (DL) models

The following three DL models are selected due to the models' proven effectiveness in maintaining long-term memory of context, which is an important property in complex Natural Language Processing tasks such as the classification of fake news.

Bidirectional Long-Short Term Memory (Bi-LSTM)

This model often encounters low training loss as a result of the large number of words used to generate embeddings. However, this also causes the model to be more prone to overfitting. Therefore, heavy regularization was implemented to prevent overfitting. In LSTM we will have 3 gates: 1) Input gate; 2) Forget gate; 3) Output gate. Gates in LSTM are the sigmoid activation functions, as shown in the following equations:

$$f_t = \sigma_g(W_f \cdot x_t + U_f \cdot h_{t-1})$$
$$i_t = \sigma_g(W_i \cdot x_t + U_i \cdot h_{t-1})$$
$$o_t = \sigma_g(W_o \cdot x_t + U_o \cdot h_{t-1})$$
$$c_t = f_t \circ c_{t-1} + i_t \circ \sigma_c(W_f \cdot x_t + U_f \cdot h_{t-1})$$
$$h_t = o_t \circ \sigma c_t$$

The equations shown here are also relevant to the GRU model, which retains two of the three gates used (excluding the Output gate).

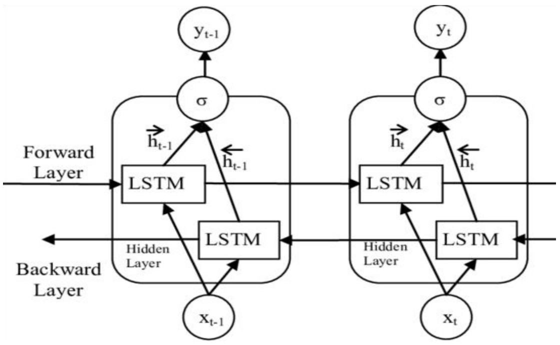

Figure 1: Bi-LSTM model units

Gated Recurrent Units (GRU)

GRU models are similar to LSTM models, but maintain fewer parameters than LSTM owing to the lack of an Output gate. To regularize the model, dropout is used in the final dense layer and recurrent layers. Further training on more epochs did not result in significant improvements to testing accuracy, and only increased training accuracy. Therefore, early stopping was employed as an additional form of regularization.

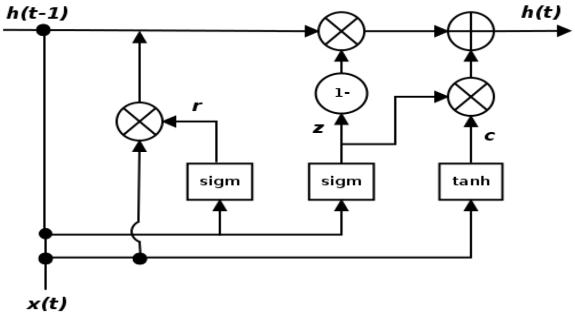

Figure 2: GRU model units

Bidirectional Encoder Representations from Transformers (BERT)

BERT is a transformer-based model developed by Google in 2018. It is the state-of-the-art model for multiple NLP tasks as of 2018 [15]. One critical hyperparameter required in the fine-tuning of BERT is the length of the input sequence. With a longer input, the model can capture more positional information. However, the attention mechanism in BERT is quadratic to sequence length, and hence the computational cost is prohibitive when input is long. Due to these limitations, an input sequence length of 512 was used, following suggested values [15].

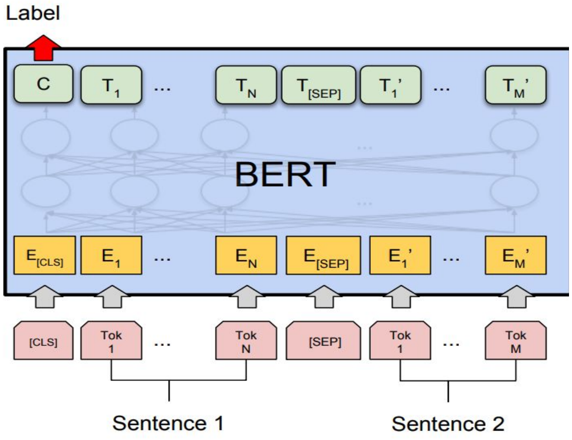

Figure 3: BERT model units

Evaluation

This section details the performance of the various models on the two datasets and analysis of the performance of the six models.

4.1 Baseline

A random guess (50% accuracy) is established as the baseline for all experiments. A model which predicts with less than 50% accuracy can have its output inverted instead.

4.2 Experimental Results

This section details model performance results on both datasets and discusses the individual performance of each model. For consistency, both datasets were split into training, validation, and testing sets at a 8:1:1 ratio. All splits were seeded with identical seeds, to ensure that the same data was used in each split, in the same order. For each model, the final optimal hyperparameter configuration is selected according to validation accuracy, and the corresponding best model is tested on the test set. Both training set and test set accuracy are detailed below.

4.2.1 Results on ISOT dataset

After extensive experiments, all models achieve accuracies higher than 95% on the test set with minimal hyperparameter tuning. The results are summarised in Table 2.

Table 2: Model Performance on ISOT dataset

Model	Train Acc	Test Acc
Logistic Regression	99.27%	98.64%
kNN	96.74%	95.61%
SVM	99.56%	99.37%
GRU	98.12%	97.23%
BiLSTM	99.03%	98.81%
BERT	**99.99%**	**99.96%**

Further analysis of the ISOT dataset was conducted, with the findings detailed in Section 5.

4.2.2 Results on FNN dataset

The performances of the models on the FNN dataset are shown in Table 3, and summarized in Figure 1. The performance of individual models will be examined in greater detail, owing to the increased variance between the models.

Table 3: Model performance on FNN dataset

Model	Train Acc (%)	Test Acc (%)
Logistic Regression	86.03	**71.48**
kNN	76.57	63.31
SVM	96.64	69.76
GRU	87.69	62.94
BiLSTM	**98.85**	61.82
BERT	97.99	71.21

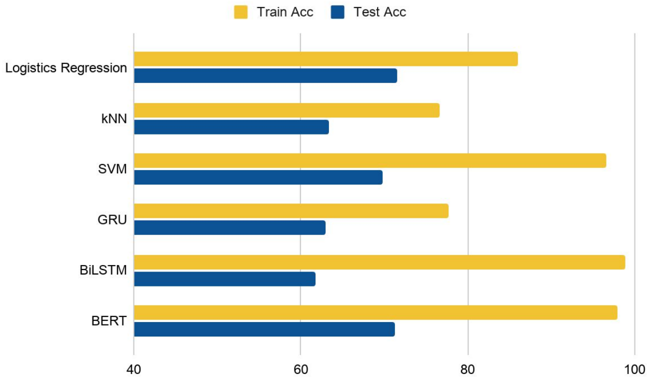

Figure 4: Accuracies of different models on FNN dataset, as %

ML models. It is observed that Logistic Regression performs notably well compared to other traditional ML methods, achieving a test accuracy of 71.28%. It does not overfit to the dataset significantly, given the comparably-small gap between training and testing accuracies. SVM, conversely, performs well in training accuracy but overfits significantly, with 69.76% test accuracy. Additionally, the training time of SVM is significantly longer than that of the other two ML models. Finally, kNN does not perform well on both training and test sets, due to the fact that it only uses distance between features to classify data points, which cannot fully capture the information in natural language sentences.

DL models. Most state-of-the-art models for NLP tasks are DL models [16]. Despite this, the testing accuracies obtained by DL models were similar to the traditional ML models. The best-performing DL model, BERT, achieved only 71.21% test accuracy, on par with Logistic Regression. Additionally, the two other DL models obtained relatively low accuracies on the test set. Although the training accuracies of these two models was noted to reach higher than 95% in other runs, the test accuracy did not consistently improve over time. Despite the use of early stopping to prevent further overfitting, the test accuracy remained approximately 60%. The large gap between training and test accuracy indicates that severe overfitting might be present. Other factors are discussed below in Section 5.

Discussion

This section seeks to analyze the results, specifically by answering the following two questions in Sections 5.1 and 5.2:

1. Why do the models perform unexpectedly well on the ISOT dataset?
2. Why do DL models fail to outperform ML models consistently?

5.1 Generalizability of models trained on each dataset

The unexpectedly-high test accuracies obtained on the ISOT dataset, combined with the unexpectedly-low test accuracies obtained on the FNN dataset, suggest a lack of generalizability between models trained on either dataset. To investigate this possibility, the ISOT dataset is further examined.

The ISOT dataset has been noted to contain some possible bias patterns [17]. These patterns include:

1. The word "Reuters" frequently appearing in real news articles as all of them are from Reuters news agency;
2. City names appearing in real news articles far more often;
3. Email and social media addresses appearing in fake news articles much more often.

These bias patterns can impact the generalizability of models trained on this dataset. To illustrate the differences between the models trained on each dataset, the BERT model (the best-performing overall model) was chosen and trained on both the ISOT and FNN datasets. Both models were then tested on both datasets, to examine the generalizability of the models to separate datasets. The train-test splits for both datasets were identical. The crosstable and overall accuracy of the models are recorded below.

Table 4: Overall accuracy of models for ISOT and FNN datasets

Overall Accuracy			
		Training Dataset	
		ISOT	FNN
Testing Dataset	ISOT	98.18%	98.22%
	FNN	49.51%	71.03%

Table 4 describes the overall accuracy of each model, depending on the datasets used to train and test the model. The overall accuracy of a model is defined as the percentage of model outputs that match the actual ground truth of the test dataset. The overall accuracy of both BERT models is very high (98%) when tested on the ISOT dataset. This indicates that the model trained on the FNN dataset was able to generalize effectively to the ISOT dataset. However, the model trained on the ISOT dataset performed extremely poorly (50% - random chance) when tested on the FNN dataset. This suggests that the features learnt from the ISOT dataset are poorly generalizable.

Table 5: Confusion matrix for ISOT and FNN datasets

Confusion Matrix (as %)						
	Training Dataset		ISOT		FNN	
Testing Dataset			Output of Model			
			Fake	Real	Fake	Real
ISOT	Ground Truth Data	Fake	52.14	0.16	52.27	0.02
		Real	1.67	46.04	1.76	45.95
FNN		Fake	49.28	0.12	36.12	13.27
		Real	50.38	0.23	15.70	34.91

Table 5 provides a confusion matrix of percentages across four categories. The cells shaded light green within the confusion matrix correspond to 'correct' answers, and the cells shaded light red correspond to 'incorrect' answers. The value in each cell indicates the percentage of overall articles that fall within that specific cell. Specifically, it can be noted that the model trained on the ISOT dataset but tested on the FNN dataset selected 'Fake' almost 100% of the time. Given that there was an approximately-equal split between real and fake news across both datasets, this resulted in a 50% accuracy for the FNN dataset, indicating poor generalizability.

These findings show that the BERT model trained on the FNN dataset is more generalizable than the model trained on the ISOT dataset. Additionally, the ISOT dataset appears to be of a lower difficulty than the FNN dataset, as evidenced by the fact that both models performed significantly better on the ISOT dataset than on the FNN dataset.

5.2 Suboptimal performance of DL methods

There is a wider gap between the training and testing accuracies of DL models compared to ML models, as discussed in Section 4.2.2. Such a gap often indicates the presence of overfitting. Considering that DL models usually have a very large number of parameters (e.g. 10^8), overfitting is likely when the training data is insufficient. Hence, the following are hypothesized:

1. GRU and LSTM perform suboptimally due to overfitting, which is caused by small dataset size.
2. BERT performs well consistently because it has been pre-trained to learn general language representations.

Due to limited timing, it is infeasible to collect new data to enlarge the dataset. Instead, the dataset can be shrunk by a ratio *r*, and the change in model performance can be observed. In addition, to verify Hypothesis 2, the BERT model can be randomly initialized and trained on the dataset. To prove these hypotheses, we should have the following observations:

1. Performances of GRU and LSTM models degrade faster than ML models due to overfitting, which is indicated by the gap between training and testing accuracy.
2. The BERT model overfits the data as well when it is randomly initialized.

The experimental data can be seen in the table below.

Table 6: Accuracies with different data sizes(r = 1/0.5/0.1)

	Logistic Regression	**KNN**	**SVM**
Train Acc	82.51/83.98/90.68	74.81/74.29/74.08	96.14/96.99/99.21
Test Acc	69.58/68.32/65.42	60.81/59.20/54.18	69.98/68.61/64.27

	GRU	**Bi-LSTM**	**BERT**
Train acc	98.28/99.23/99.60	8744/93.99/100.00	97.98/98.18/96.75
Test acc	58.44/59.49/59.37	0.6321/0.5793/0.5591	71.08/67.46/67.47

When *r* decreases from 1 to 0.5, the accuracy gap of GRU and LSTM models increases significantly, compared to ML methods. Similar observations can be found when *r* decreases from 0.5 to 0.1. This is consistent with Observation 1. In addition, when BERT is randomly initialized and trained following the training procedure in the original paper [15], optimization is found to be very difficult, i.e. both the training and testing accuracy are not significantly greater than 50%. This is possibly due to the lack of computational power needed for training the BERT model from scratch. However, although Observation 2 cannot be made directly, it has been demonstrated that pre-training performs a vital role in helping BERT fit to new data. Hence, we can conclude that Hypothesis 1 is correct, while Hypothesis 2 is likely correct.

Conclusion

This project aimed to determine the feasibility of the selected models in accurately classifying fake news articles. It was discovered that excellent accuracy was attainable on some testing datasets (e.g. ISOT), but not on others (e.g. FNN), likely due to differences in the inherent difficulty of the dataset. It is also noted that models trained solely on the ISOT dataset without significant data cleaning are likely not to be generalizable to other datasets, and recommend training models on more than just the ISOT dataset.

This project is limited in the sophistication of the DL models used. Existing literature suggests that the DL models used should achieve superior performance to the traditional ML models used, but this trend was not corroborated by the data. The discrepancy in results is possibly due to suboptimal hyperparameter choices stemming from a lack of computational resources, or an insufficiently-complex model (as mentioned in the evaluation of BERT in Section 3.5.2). With state-of-the-art models, it is feasible to train a DL model to accurately classify fake news articles. This paper concludes that despite the results obtained, there is reason to claim that the accurate classification of fake news articles is feasible.

Moving forward, we will be looking at how we can implement a fake news detector in a more accessible medium, such as a web extension that can warn the user when they visit a site with potentially fraudulent news.

Acknowledgements

Team 20 would like to sincerely extend our gratitude to our lecturer, Prof. Kan Min-Yen, and our TA, Shreyas, for their helpful and timely feedback. Their guidance was essential in honing our direction throughout this project.

Project Repository

The code used in this project can be located in the repository stored at https://github.com/BoAi01/CS3244-fake-news-detection.

References

[1] Cambridge Dictionary. Retrieved at https://dictionary.cambridge.org/dictionary/english/fake-news

[2] Shao C., Ciampaglia G. L., Varol O., Yang K. C., Flammini A., and Menczer F. (2018). The Spread of Low-credibility Content by Social Bots

[3] Horne, B. D. and Adali, S. (2017). This just in: Fake news packs a lot in title, uses simpler, repetitive content in text body, more similar to satire than real news. arXiv preprint arXiv:1703.09398.

[4]Shu K., Sliva A., Wang S., Tang J., and Liu H. (2017). Fake news detection on social media: A data mining perspective. ACM SIGKDD explorations newsletter, 19, pp. 22-36

[5]Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. arXiv preprint arXiv: 1301.3781.

[6]Ghosh S. and Shah C. (2018). Towards automatic fake news classification, Proceedings of the Association for Information Science and Technology, 55, pp. 805-807

[7]Ruchansky N., Seo S., Liu Y. (2017). Csi: A hybrid deep model for fake news detection. Proceedings of the 2017 ACM on conference on information and knowledge management, ACM , pp. 797-806

[8]Ma J., Gao W., Mitra P., Kwon S., Jansen B., Wong K., and Cha M. (2016). Detecting Rumors from Microblogs with Recurrent Neural Networks.. In IJCAI. 3818–3824.

[9]Araque O., Corcuera-Platas I.,Sanchez-Rada J.F.,Iglesias, (2017). C.A. Enhancing deep learning sentiment analysis with ensemble techniques in social applications. Expert Systems with Applications, 77, pp. 236-246

[10] Ahmed, H., Traore, I., Saad, S. (2018) Detecting opinion spams and fake news using text classification. J. Secur. Priv. 1(1), e9

[11]Amir J. B. (2020) Retrieved at https://ieee-dataport.org/open-access/fnid-fake-news-inference-dataset

[12]Goldani M. H., Momtazi S., and Safabakhsh R. (2020). Detecting fake news with capsule neural networks. arXiv preprint arXiv:2002.01030

[13]Ahmed, H., Traore, I., & Saad, S. (2017). Detection of online fake news using N-gram analysis and machine learning techniques. In International Conference on Intelligent, Secure, and Dependable Systems in Distributed and Cloud Environments (pp. 127–138). Springer.

[14]Vishwakarma D.K., Varshney D., Yadav A. (2019). Detection and veracity analysis of fake news via scrapping and authenticating the web search. Cognitive Systems Research, 58, pp. 217-229

[15]Devlin J., Chang M. W., Lee K., Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding

[16] Bishop and Christopher M. (2006) Pattern Recognition and Machine Learning.

[17] Sebastian K., ChoraśRafał K., and Paweł K. W. (2020). Sentiment Analysis for Fake News Detection by Means of Neural Networks

Image References

Figure 1: https://www.scirp.org/html/3-7800645_99760.htm

Figure 2: https://www.researchgate.net/figure/Structure-diagram-of-the-gated-recurrent-unit-GRU-Structure-diagram-of-the-gated_fig2_327710626

Figure 3: https://nlp.stanford.edu/seminar/details/jdevlin.pdf