Fake News Detection

Natural Language Processing Exam 2021

Emil Trenckner Jessen
AU ID: au604547

Johan Kresten Horsmans
AU ID: au618771

School of Communication and Culture,
Aarhus University
Langelandsgade 139, 8000, Aarhus C, Denmark

December 22nd, 2021

Keywords: NLP, Machine Learning, Fake News Detection, English, BERT, Dynamic Word Embeddings.

Character count: 26,515

Regarding initials:

ETJ stands for *Emil Trenckner Jessen*.
JKH stands for *Johan Kresten Horsmans*.

Each segment of this assignment has been marked with respect to who has been the primary contributor. Nonetheless, it is essential to note that the entire exam is considered a completely collaborative effort since all sections have been written and edited extensively by both authors in tandem. Furthermore, the coding and analysis has been carried out jointly by both authors.

The code for our analysis can be found in the following GitHub repository.

If one wishes to replicate our analysis, we have created a thorough step-by-step guide in the README under the section *usage*. The repository includes a bash script that automatically downloads the datasets that were utilized. To do this, one first needs to clone the github repository linked above. To acquire the repository and data, please run the following code in a unix-based bash terminal:

NOTE: This repository is designed to work on a *JupyterNotebook LaTeX* instance on *UCloud*. If you wish to replicate the analysis, slight variations may be necessary depending on your machine.

```
git clone https://github.com/emiltj/NLP_exam_2021
cd NLP_exam_2021
pip install -r requirements.txt
bash download_data.sh
```

You are now ready to run the code.

Contents

1 Introduction (JKH)

In 2016, the former Pakistani defense minister, *Khawaja Muhammad Asif*, posted an ominous message on his official Twitter account after reading an article on a nuclear threat from the Israeli minister of defense. The message from Asif was a reminder to Israel: "[...] *Israel forgets Pakistan is a Nuclear state too.*" (Asif, 2016, as cited in Goldman, 2016). However, shortly after, the news article that Asif had replied to was identified as a fabricated fake (Goldman, 2016; Graham-Harrison, 2016). Such fake news articles have been found not just to affect political decision-makers but also the public opinion on political matters pertaining to elections, foreign- and domestic policies, and more general issues such as vaccinations and COVID-19 (Allcott & Gentzkow, 2017; Grinberg et al., 2019; Grinberg, 2019; Orso et al., 2020). Fake news have been around for decades, but a recent rapid increase in misinformation on social media has made this an increasingly problematic tendency (Burkhardt, 2017). Therefore, we argue that the need for courses of action to limit the impact of fake news is apparent. However, given the amount of content shared on the internet every day and the fact that the majority of fake news articles are published on small platforms, the task at hand would be insurmountable if the articles were to be evaluated manually.

Given this difficulty in evaluating and detecting fake news in the large stream of digital content, many studies have attempted to automate the detection process using Machine Learning (*ML*) approaches (Ahmed et al., 2018, 2017; Horne & Adali, 2017; Pérez-Rosas et al., 2017; Reis et al., 2019). Although such ML studies have achieved accuracies of up to 92% (Ahmed, 2017), very little work has been done to assess the ecological validity of these reported performances. In other words, researchers may be able to correctly classify and detect fake news within their own dataset, but will these performances generalize to datasets created from different sources? Assessing such generalizability of the models created within the field is currently not possible since there, to the authors' knowledge, are no publicly available models fine-tuned for the task.

Furthermore, we hypothesize that another problem with the task of detecting fake news could be that the content of the articles may change over time. A classifier trained in 2018 would probably not be very sensitive towards fake news about COVID-19 since such articles did not exist at the point in time where the employed training dataset was generated. In other words, the topics handled in fake news change over time, and model predictions can therefore become over– or under-sensitive to specific keywords that do not necessarily correspond to fake news in general, but rather to fake news within a certain time period. As such, we argue that the performance of a given fake news detection classifier would most likely decrease over time due to the non-staticity of fake news content.

Upon inspecting the field of ML classification of fake news, in context of the problems raised above, we deem it relevant to conduct further research within this realm. Followingly, the main scope of this assignment consists of exploring and assessing these problems. More specifically, the problems of interest are:

1. *The potential lack of generalizability between datasets.*
2. *The non-staticity of news articles.*

To address the first problem, we train and cross-test BERT models on two very different datasets, each used in published studies; one large dataset of relatively poor quality and another containing significantly fewer entries but of higher quality (for more info, see *section 2.1.1*). We do this to assess whether our hypothesized lack of generalizability between datasets for fake news ML-models holds true.

To address the second problem, we investigate the non-staticity of fake news data by exploring how the word embeddings in fake articles change over time as a result of varying news topics dominating the media.

Solving these problems exhaustively would be beyond the scope of this assignment. Therefore, our focus is instead to direct attention towards the raised issues and propose guidelines for future development and research within the field.

2 Classification

2.1 Methods

2.1.1 Data sources (ETJ)

The first dataset (*dataset 1*) utilized in our analysis was acquired through the *University of Victoria*'s research laboratory: *Information security and object technology* (ISOT). The dataset contains a total of 44,898 articles (hereof 21,417 real and 23,481 fake) from 2015 to 2017 and has been constructed by courtesy of Ahmed et al. (2017, 2018). The real news articles have been collected from the news website *reuters.com*, while the fake news articles were scraped from an assembly of sources that were flagged as unreliable by either *Politifact*, *Wikipedia* or both. For more information, please refer to the original papers (Ahmed et al., 2017, 2018).

The second dataset (*dataset 2*) is a combination of two sub-datasets. One was acquired by Horne et al. (2017) in the time period between 2016 and 2017, and the other was put together by Silverman (2016), who collected data in the time period between February and August 2016. The combination of the two sub-

datasets is publicly available through a GitHub repository by Horne et al. (2017). Dataset 2 contains a total of 248 fake– and real news articles (counting 120 and 128, respectively). The sub-dataset by Horne & Adali (2017) furthermore contains 75 satirical news articles, while the sub-dataset by Silverman (2016) contains three satirical articles. None of the satirical news articles were included in this study. The majority of real news articles were from *Business Insider's* "*Most Trusted*" list (Engel, 2014) and were collected from the following sources: *Wall Street Journal*, *The Economist*, *BBC*, *NPR*, *ABC*, *CBS*, *USA Today*, *The Guardian*, *NBC*, and *The Washington Post*. Most of the fake news entries came from journals and magazines found in Zimdars "*Fake News*" list (Zimdars, 2016). The journals and magazines in question are: *Ending The Fed*, *True Pundit*, *abcnews.com.co*, *DC Gazette*, *Liberty Writers News*, *Before its News*, *InfoWars* and *Real News Right Now*. For more information on dataset 2, please refer to the paper by Horne et al. (2017).

2.1.2 Data quality assessment (ETJ)

Although both datasets had been reported to have been preprocessed by the original authors, we carried out a manual screening to identify any potentially unwanted information in the data. This inspection ensured that any necessary additional preprocessing steps – apart from fundamental text preprocessing – could be carried out. Upon inspecting the datasets, no further preprocessing proved necessary for dataset 2. However, several problematic issues pertaining to the data acquisition were identified in the large dataset 1 by Ahmed et al. (2017, 2018). In this dataset, most articles labelled real news contained the words "*Reuters*" and "*verified*" along with city names in the beginning of the text. Similarly, long identical sequences of text were consistently included in the beginning and end of most fake news entries. These sentences were entirely unrelated to the article and seemed to be artefacts from the automatic scraping process. Furthermore, the inspection revealed that dataset 1 included duplicates, links, hashtags, Twitter handles, and more than 600 empty entries.

2.1.3 Preprocessing (JKH)

The issues identified in dataset 1 could potentially cause models to learn non-generalizable patterns for classification if not dealt with appropriately. To account for this problem, the following preprocessing was carried out:

1. First, we removed entries that were either empty or only contained whitespaces.
2. Followingly, we removed duplicate entries in the dataset.
3. Regular expressions were employed to remove words and sequences unrelated to the content of the articles. This included:
   (a) *[city name] Reuters -*

(b) *The following statement [... everything up to ...] accuracy*

(c) pic.twitter.com/

4. Furthermore, we removed the following elements since they were only present in the fake-news data.

   (a) *Hashtags* (e.g. #NotMyPresident)

   (b) *Twitter tags* (i.e. @[username])

   (c) *([CAPSLOCKED WORD])* (e.g. (VIDEO))

5. We proceeded to remove punctuation.

6. All text was transformed to lowercase.

7. We used NLTK (Bird, 2006) to remove stopwords.

8. We then lemmatized the corpus with respect to part-of-speech tags.

9. Followingly, based on manual inspection, we found new systematic patterns that were only in the fake data. These patterns were removed to avoid systematic bias in the data:

   (a) *21st century wire say*

   (b) *21st century wire*

   (c) *filessupport [... to end of entry]*

   (d) *21wire*

Subsequently, we split both dataset 1 and dataset 2 into unique training–, validation– and testing datasets with a proportionally equal amount of fake– and real news. The testing datasets were constructed with 20% of the comments in each respective dataset. The validation was made with 10% of the remaining training data. For the specific implementation of the preprocessing, please refer to the `Analysis.ipynb` -notebook.

2.1.4 Modeling (JKH)

For the modeling, we utilized the *BERT Base (uncased)* model from Hugging Face (Devlin et al., 2019). This was implemented in Python with the *simple transformers* framework (Rajapakse, 2021). We proceeded to train two BERT models – one on each dataset. Using the validation dataset, we optimized the hyperparameters based on informed trial and error (see *table 1*).

Parameter	Value
Use CUDA:	*False*
Reprocess input data:	*True*
Epochs:	*Dataset 1: 3* *Dataset 2: 15*
Max sequence length:	*512*
Batch size:	*Dataset 1: 128* *Dataset 2: 16*
Learning rate:	*1e-5*

Table 1: Specified hyperparameters for BERT-model training. Non-specified parameters remained default.

The models were trained on UCloud using a virtual machine with 64 CPU cores and 376 GB RAM. We did not have a GPU available and, therefore, we did not utilize CUDA. This is also the reason for utilizing a high batch size and a relatively low amount of epochs for the model trained on dataset 1. We did this since training took approximately 2 hours per epoch for this dataset. Nonetheless, we deemed that training was sufficient based on excellent results on the validation dataset. In sum, our analysis pipeline can be found in *figure 1*.

2.2 Results (JKH)

	Trained on dataset 1	**Trained on dataset 2**
Evaluated on dataset 1	1.00	0.59
Evaluated on dataset 2	0.58	0.80

Table 2: Macro F1-scores for the BERT-classification models.

2.3 Discussion (ETJ)

The model trained on dataset 1 achieved a macro F1-score of 1.00 when tested within-dataset on the test partition. However, the model only attained a score of 0.58 when tested out-of-corpus. Similarly, the model trained on dataset 2 saw a substantial performance drop, going from a macro F1-score of 0.80 to a score of 0.59, when changing the testing data from within– to across datasets. In sum, model performances seem to drop considerably when tested on new data, suggesting that the models lack generalizability. We argue that the low level of generalizability must stem from a large degree of heterogeneity between the datasets since each model only performs well on the dataset that it was fine-tuned upon. If the data used for training is not

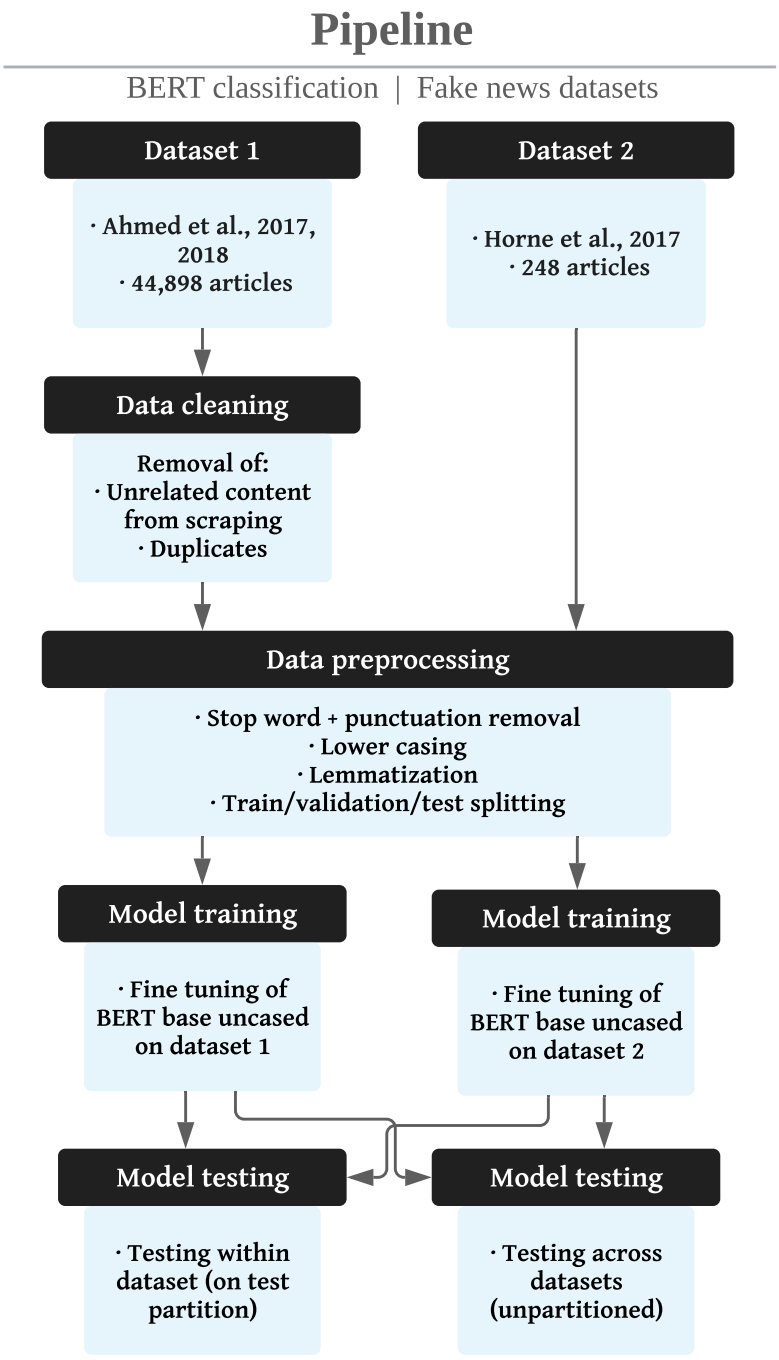

Figure 1: Visualization showing the classification pipeline. The two datasets went through the same process, although dataset 1 went through additional cleaning prior to the regular preprocessing.

representative of the overarching task at hand, then the model will perform suboptimally on other datasets.

Since our models were trained on datasets that have already been utilized in published research, we propose that more work ought to be carried out to assess the generalizability of new models. Moreover, although scientific studies report to have constructed highly discriminatory models with seemingly high performances, our findings suggest that the capabilities of such models – if actually applied to real-world tasks – would be substantially inferior to what would have been expected if one was to take the reported performance scores at face value.

However, the low generalizability of models – resulting from heterogeneity between the two datasets – does not explain how the model trained on dataset 1 achieved a macro F1-score of 1. The high performance brings about an important point of critique for dataset 1; The labels must be exceedingly easily differentiable, even after conducting additional data cleaning. Upon further manual inspection, the fake news entries were found to contain large amounts of abbreviations, slang and misspellings – features that were not present to the same extent for the real news entries. The acquisition process for dataset 1 can further be critiqued as the authors of the dataset assumed that sources flagged by Politifact or Wikipedia exclusively published fake news articles. Through manual inspection, it appears as if some articles do not seem to be fake news but rather accounts of non-verified opinions. Therefore, it does not seem unlikely that the flagged websites also produce truthful content on occasions. Finally, Politifact does not explicitly state their flagging procedure, which further affords skepticism.

Regarding future work within the field, we have some recommendations based on our classification findings across datasets. The low generalizability of the models and the critique of the quality of dataset 1 both point in the same direction: The construction of large, representative datasets of high quality is of paramount importance if we are to utilize and implement ML models for fake news detection.

3 Dynamic word embedding analysis

3.1 Analysis (ETJ)

The term *word embeddings* refers to vector representations of words in an embedding space wherein words with similar meanings appear closer together. Distances between words in any n-dimensional embedding space thus represent semantic similarity and may be calculated using euclidean-, cosine-, or other distance formulas (Jurafsky & Martin, 2001). For this analysis, cosine distance is utilized as distance metric. Following the formula below, distances may therefore range from 0 (perfectly similar) to 2 (perfectly dissimilar).

$$cosinedistance = 1 - cos(\theta) = 1 - \frac{\sum_{i=1}^{n} a_i b_i}{\sqrt{\sum_{i=1}^{n} a_i^2}\sqrt{\sum_{i=1}^{n} a_i^2}}$$

where:

- $cos()$ = cosine function
- θ = angle in degrees between two vectors in n-dimensional embedding space
- a = word embedding vector for word a
- b = word embedding vector for word b

For our dynamic word embedding analysis, we utilized the fake news from dataset 1. This consisted of articles from 1,006 days, loaded into Python as a pandas dataframe. Firstly, we transformed the date representation to be consistent across the entire dataset. Followingly, we segmented the dataset into five temporal periods, each corresponding to 20% of the total time span of the dataset. The time periods were the following:

1. *March 31, 2015 - October 18, 2015.*
2. *October 19, 2015 - May 6, 2016.*
3. *May 7, 2016 - November 23, 2016.*
4. *November 24, 2016 - June 12, 2017.*
5. *June 13, 2017 - December 31, 2017.*

Preprocessing of the data was subsequently carried out following the procedure described in section 2.1.3 *Preprocessing*. Lastly, we created five concatenated .txt-files, each containing all the articles from the respective periods. For an exemplification of how this was implemented in Python, please see the *periods* section in the `Analysis.ipynb`-notebook.

For the word-embedding analysis, we based our modeling on the resources made available by Barzokas et al. (2020). Their scripts were publically available on GitHub with an MIT license granting permission to modify and utilize the code free of charge as long as the licensing is not changed in the modified version. Their processing was based on the fastText-framework by Bojanowski et al. (2017) and consequently, so was our analysis. Multiple modifications of the code were implemented to ensure it fit our project. First and foremost, their analysis was based on Greek text-corpora. As such, we had to alter the script to work

on English texts. Furthermore, the script was designed to download web scraped resources and transform them into the format required for running the scripts. Since we already had our dataset available, we used our own scripts to transform the dataset into the desired format. Moreover, although the original script was designed to find the words whose embeddings had the highest cosine distance between periods, we had to modify the script to save the raw cosine distances into a .txt-file. We also removed various data-cleaning steps they had encoded into their scripts. Lastly, we changed the parameters of the fastText model to fit our data most optimally.

For our analysis we used the fastText skipgram model with the following hyperparameters:

Parameter	Value
Model:	*skipgram*
Context window size:	*20*
Epochs:	*5*
Word embedding size:	*100*
Minimum word occurences:	*50*
Minimum char n-gram length:	*3*
Maximum char n-gram length:	*6*
Number of negative samples:	*5*
Number of CPU threads:	*12*

Table 3: Parameters for the fastText word embedding model. Non-specified parameters remained default.

3.2 Findings (JKH)

After training the model, we calculated the words with the highest internal cosine distances between the first– and the last period. This was done to identify words that would best illustrate the potential non-staticity of fake news word embeddings. We decided to focus on nouns since we argue that they are more prone to appear in different contexts compared to other categories of words. This selection of words with the highest semantic shift across the time periods can be found in *table 4*.

Previous studies show that cosine distances above 0.8 imply quite significant semantic shifts for a word (Bamler & Mandt, 2017; Barzokas et al., 2020). In the light of this, it is apparent that there is a notable semantic shift for the words reported in *table 4*. This is a general trend for the dataset where we found 925 words with a cosine distance above 0.8. We argue that this supports our hypothesis pertaining to the

impermanence of word embeddings over time for fake news data.

Word	Cosine distance
jail:	*1.189*
eric:	*1.175*
president:	*1.170*
clinton:	*1.168*
graham:	*1.132*
donald:	*1.126*
russia:	*1.018*
investigator:	*0.998*
voter:	*0.990*

Table 4: Cosine distances of selected words between the first– and last period. Rounded to three decimals.

To investigate what might be driving such a semantic shift, we decided to explore the nearest neighbours for the word *russia*, which had one of the highest internal cosine distance scores (*1.018*). We computed the ten nearest neighbours for the word across each of the five time-periods. If any two words shared semantic meaning, we excluded the one with the lowest cosine distance (i.e., excluding *russians* or *sanctioning* if *russian* or *sanction* also appeared). These results can be found in *figure 2*. Here, we see an example of how real-world events can drastically change the word embedding space. Across the first three periods, we find little variation in the list of nearest neighbours where most of the words are related to matters of military (e.g., *military*, *region*, & *air*) and political affairs (e.g., *syria*, *diplomatic*, & *sanction)*. Suddenly, in the fourth– and fifth periods, we see an array of words relating to, seemingly, quite different matters. We argue that this is most likely a product of the investigation of Russian interference in the 2016-election, which took place during these time periods (Mueller & Cat, 2019). Some of the keywords indicating this are: *collude*, *investigate*, *interfere*, *evidence*, *russiagate* and *mueller* (the director of FBI during the investigation).

As such, we argue that our findings showcase the problem of how non-staticity interferes with building word-embedding representations that are reliable and generalizable across time periods.

For future prospects within the field, we, therefore, recommend that models need to be regularly re-fine-tuned on continuously updated datasets, reflecting the contemporary media landscape in the world. This may seem like a tedious task that defeats the purpose of an automatized classification model, but we argue that it would still be less cumbersome and time-consuming to generate such datasets compared to manually filtering all news articles posted online.

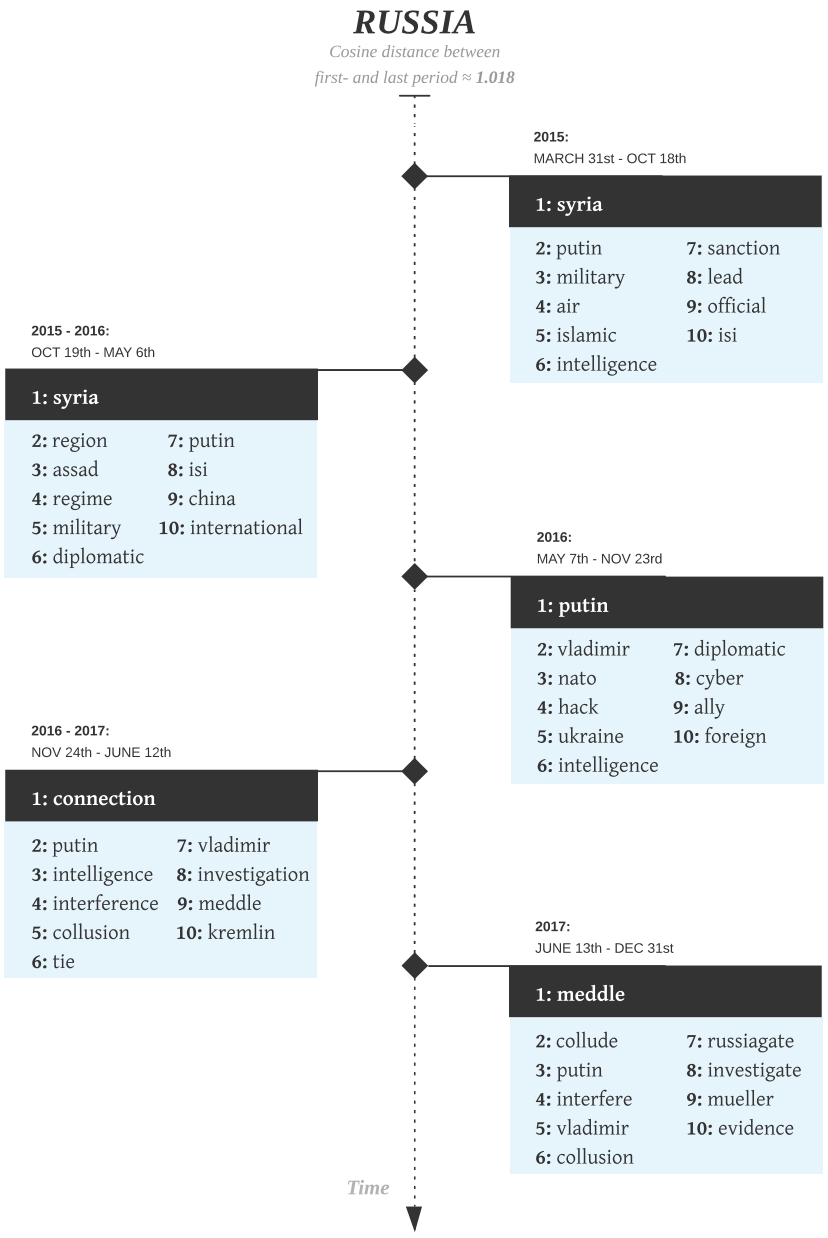

Figure 2: Visualization showing the temporal evolution of nearest neighbours in word embedding space for the word "russia".

4 Conclusion (ETJ)

We have hypothesized and explored two potential pitfalls within the field of automatized fake news detection. These raised problems pertained to *the potential lack of generalizability between datasets* and *the non-staticity of news articles*. To address the first issue, we built two BERT-classification models, which were trained and cross-tested on two separate datasets that had been utilized in previously published studies. Here, we found that the BERT models achieved high performance on the dataset it was trained on but had a very poor generalizability to the other dataset. Furthermore, we highlighted issues found in one of the datasets which resulted in unrealistically high performance metrics. For future research within the field, we, therefore, argue that the creation of high-quality representative datasets need to be developed if one hopes to build a reliable classifier. To address the second problem, we created a dynamic word embedding model which calculated how the word semantics of our largest fake news dataset changed across time periods, both within– and between words. Here we found large semantic shifts for a wide array of words. We argue that this is most likely a product of how the content of the media landscape changes as a function of time. For future improvements within the field, we, therefore, recommend that models are continually re-fine-tuned on continuously updated datasets that reflect the contemporary news stream.

5 References

Ahmed, H., Traore, I., Saad, S. (2018). Detecting opinion spams and fake news using text classification. *Security and Privacy, 1*(1), e9.

Ahmed, H., Traore, I., Saad, S. (2017). Detection of Online Fake News Using N-Gram Analysis and Machine Learning Techniques. In I. Traore, I. Woungang, A. Awad (Eds.), *Intelligent, Secure, and Dependable Systems in Distributed and Cloud Environments* (pp. 127–138). Springer International Publishing. https://doi.org/10.1007/978-3-319-69155-8$_9$

Allcott, H., Gentzkow, M. (2017). Social media and fake news in the 2016 election. *Journal of Economic Perspectives, 31*(2), 211–236.

Bamler, R., Mandt, S. (2017). Dynamic word embeddings. *International Conference on Machine Learning*, 380–389.

Barzokas, V., Papagiannopoulou, E., Tsoumakas, G. (2020). Studying the Evolution of Greek Words via Word Embeddings. 1*1th Hellenic Conference on Artificial Intelligence*, 118–124. https://doi.org/10.1145/3411408.3411425

Bird, S. (2006). NLTK: The natural language toolkit. *Proceedings of the COLING/ACL 2006 Interactive Presentation Sessions*, 69–72.

Bojanowski, P., Grave, E., Joulin, A., Mikolov, T. (2017). Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics, 5*, 135–146.

Burkhardt, J. M. (2017). *Combating fake news in the digital age* (Vol. 53). American Library Association.

Devlin, J., Chang, M.-W., Lee, K., Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *ArXiv:1810.04805 [Cs]*. http://arxiv.org/abs/1810.04805

Engel, P. (2014). Here are the most-and least-trusted news outlets in America. *Business Insider, 21.*

Goldman, R. (2016, December 24). Reading Fake News, Pakistani Minister Directs Nuclear Threat at Israel. *The New York Times.* https://www.nytimes.com/2016/12/24/world/asia/pakistan-israel-khawaja-asif-fake-news-nuclear.html

Graham-Harrison, E. (2016). Fake news story prompts Pakistan to issue nuclear warning to Israel. *The Guardian.* https://www.theguardian.com/world/2016/dec/26/fake-news-story-prompts-pakistan-to-issue-nuclear-warning-to-israel

Grinberg, N., Joseph, K., Friedland, L., Swire-Thompson, B., Lazer, D. (2019). Fake news on Twitter during the 2016 US presidential election. *Science, 363*(6425), 374–378.

Horne, B., 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. Pro*ceedings of the International AAAI Conference on Web and Social Media, 11*(1).

Jurafsky, D., Martin, J. H. (2021). *Speech and Language Processing 3rd ed. Draft.* https://web.stanford.edu/ jurafsky/slp3/

Mueller, R. S., Cat, M. W. A. (2019). *Report on the investigation into Russian interference in the 2016 presidential election.* US Department of Justice Washington, DC.

Orso, D., Federici, N., Copetti, R., Vetrugno, L., Bove, T. (2020). Infodemic and the spread of fake news in the COVID-19-era. *European Journal of Emergency Medicine*, 10.1097/MEJ.0000000000000713. https://doi.org/10.1097/MEJ.0000000000000713

Pérez-Rosas, V., Kleinberg, B., Lefevre, A., Mihalcea, R. (2017). Automatic Detection of Fake News. *ArXiv:1708.07104 [Cs].* http://arxiv.org/abs/1708.07104

Rajapakse, T. (2021). *Simple Transformers* [Python]. https://github.com/ThilinaRajapakse/simpletransformers (Original work published 2019)

Reis, J. C. S., Correia, A., Murai, F., Veloso, A., Benevenuto, F. (2019). Supervised Learning for Fake News Detection. *IEEE Intelligent Systems, 34*(2), 76–81. https://doi.org/10.1109/MIS.2019.2899143

Silverman, C. (2016). This analysis shows how viral fake election news stories outperformed real news on Facebook. *BuzzFeed News*, 16.

Zimdars, M. (2016). False, misleading, clickbait-y, and satirical "news" sources. *Google Docs*.