Proceedings of the Federated Conference on Computer Science and Information Systems pp. 489–495
DOI: 10.15439/2018F110
ISSN 2300-5963 ACSIS, Vol. 15

Evaluating Combinations of Classification Algorithms and Paragraph Vectors for News Article Classification

Johannes Lindén, Stefan Forsström, Tingting Zhang
Departement of Information Systems and Technology
Mid Sweden University
851 70 Sundsvall, Sweden
Email: johannes.linden@miun.se, stefan.forsstrom@miun.se, tingting.zhang@miun.se

***Abstract*—News companies have a need to automate and make the process of writing about popular and new events more effective. Current technologies involve robotic programs that fill in values in templates and website listeners that notify editors when changes are made so that the editor can read up on the source change on the actual website. Editors can provide news faster and better if directly provided with abstracts of the external sources and categorical meta-data that supports what the text is about. In this article, the focus is on the importance of evaluating critical parameter modifications of the four classification algorithms Decisiontree, Randomforest, Multi Layer perceptron and Long-Short-Term-Memory in a combination with the paragraph vector algorithms Distributed Memory and Distributed Bag of Words, with an aim to categorise news articles. The result shows that Decisiontree and Multi Layer perceptron are stable within a short interval, while Randomforest is more dependent on the parameters best split and number of trees. The most accurate model is Long-Short-Term-Memory model that achieves an accuracy of 71%.**

I. Introuction

THERE are several approaches to extracting the key points of non-formatted text to be able to retell the most important information to the reader. A common problem is the over all descriptive word of the text, such as this text is about Kultural arts. In this article we will call this information a category. Other problems involve retrieving shorter summaries of text documents and computing other meta data describing the text content. The purpose is to make the text more available to readers/writers, and from there link the text the appropriate audience by for example personalization and document search algorithms.

Swedish journalists categorize their news articles manually. It is a time-consuming task and yields inconsistent results. A previous study by Oscar Hjelmstedt and Mats Sellfors shows that journalists needs to take advantage of algorithms that can manage news content to get a better understanding of how they work and move on from old habits of news paper press that are very different from digital media [1]. In the future, news will to a greater extent be written by using deep learning algorithms to write news faster and at a lower cost [1]. It is therefore important that the journalists have an understanding and know how to work with the new working conditions. Hence, this research seeks to answer the following research questions:

1) Will a combination of classification and paragraph vector algorithms improve the results of the categorizations?
2) To which extent and which combinations of classification and paragraph vector algorithms shows the best accuracy for new articles?

In this article, the focus will be the paragraph vectors distributed memory and distributed bag of words as described by Mikolov et al. [2]. As an additional layer of algorithms we categorize the paragraph vector using the standard data mining algorithms: decision tree, random forest and multi layer perceptron. A comparison between the result of our work and other categorization algorithms like Fasttext and Lai Siwei et al.'s classification algorithm will be presented and the f-score metric will be evaluated [3], [4].

A. Outline

This article will first go through some related work that already have been done in the research field. Secondly a approach/models description about the algorithms used and how they are combined in the experiments conducted described in the next section. The scores that are based on the model evaluation experiments are then presented in the result section, followed by the discussion and conclusions of the project. Finally suggestions for future work are presented.

II. Related Work

In the field of text categorization, there is already existing research that should be considered. Facebook announced their own categorization algorithm called fasttext a few years ago, which shows good performance in speed [3]. In a matter of seconds, a trained fasttext model is ready to categorize texts in comparison to other algorithms like Gensim with the same dataset setup this is fast, there by the name. The reason for the speed increase is most likely their n-gram implementation that mainly introduces good results for the syntactical parts of the text, but weakens the semantic parts. The fasttext algorithm gets a way with less computational complexities and still performing well on syntactic problems [4]. In this article

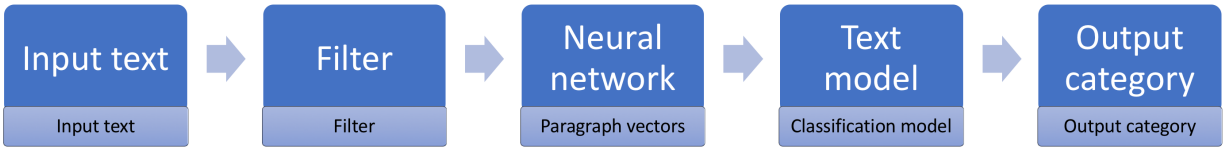

Fig. 1. An overview of the approach and model

several steps to categorize the text are performed. An LSTM classification algorithm using a multi-timescale approach to prolong the long-term memory in the network is proposed by Liu et al [5]. Liu uses English data sources with question-answer and binary data.

Standford University conducts research in the field of natural language processing, which includes text categorization of Twitter feeds. They consider a large number of Twitter posts and categorizes the posts into positive and negative. They also consider emoticons noisy labels. Although they are showing good results they categorize the posts of two categories and emoticons are quite unstable facit. The Standford work is in line with this article in that it too uses predefined categories for training, but positive and negative categories are more abstract which implies human error is greater. [6]

A completely different approach to text categorization is term association, associates certain rules and patterns of the text with certain categories. This approach relies on the pruning phase of the model, since a lot of fake rules will emerge from a large dataset. Therefore Maria-Luiza Antonie and Osmar R. Zaïane [7] came up with three pruning rules that reduce the amount of rules and increases the accuracy of the model in 2002. The paper shows that an association model for categorization can be both accurate and fast.

The algorithms used when dealing with natural language processing are commonly also used in image processing. For example a paper about practical study of network image based classification by Dabrowski, Marek et al uses a convolutional network to categorize images which can be compared to a very deep convolutional network approach to character based language classification by Conneau Alexis et al [8], [9]. These algorithms depending on the dataset takes often long time to converege given the initial weights the time of the result could vary between different training runs Polap, Dawid et al shows one method to use multi-threaded learning with a multi-core solution to achieve faster training time [10].

Google released a data source platform called GDELT that stores a lot of news metadata from all over the world. The system has the computer power to store and monitor world news on the internet from certain news sources, new events as well as events reaching as far back in time as 1979. Over 200 million events are recorded from over 240 countries and available for live requests. In 2013, a comparison between the GDELT and ICEWS was made that compared the popularity and scale of the two data sources. [11], [12]

Other competitive algorithms that provide a document vector for a given text are LDA algorithms and text ranking algorithms. In an article by Thanda et al. [13] they compare the different algorithms in a systematic matter to find relations between math queries.

III. Approach and Model

We propose a four step model that predicts categories of arbitrary text paragraphs. See Figure 1 for an overview of our implementation. The input in Figure 1 is the algorithm parameters θ and a single document D, which is interpreted as a sequence of words $w_1, w_2, \cdots, w_n$. The output of the model is a set of category probabilities $c_1, c_2, \cdots, c_i, \cdots, c_m$ where

$$c_i = P\left(\text{ith category}|D, \theta\right) \approx P\left(\text{ith category}|D\right) \quad (1)$$

. Before the actual training, the data is filtered from text paragraphs that only consists of a link to another article and that does not represent any categorical value. The combination of step three and four is the machine learning part, which will answer the research questions. The algorithms used are described in the following sections.

A. Input Text

The input of the proposed algorithm is an unstructured sequence of words forming a text paragraph. The text should be in Swedish and can be of any length. Although in this article the tested the text sizes have a length between 5 to 600 words.

B. Input Filter

Before the text can serve as the input to the model the text needs to be filtered to remove special characters. Exclamation marks and question marks are replaced by full stops. Commas, references and document links are removed. The purpose of the filter layer is to make the paragraph uniform, so that the model can be processed with as few exceptions as possible. In this step, one scenario was to filter on verbs and nouns to make the input data more precise to the point and thus describe the category using narrow information without noise words. To filter on these words a part-of-speech tagger was used.

Part-of-speech taggers (POS-tagger) are used to extract the sentence structure in the form of a dependency tree and the cooresponding word's tags [14]. A tag indicates if the word is a verb, noun, preposition or any other type. The dependency tree has a root word node and child words that directly relates to the parent word, an example is shown in Algorithm III-B. Google released a POS-tagger called SyntaxNet with state-of-the-art performance, and one year later announced an

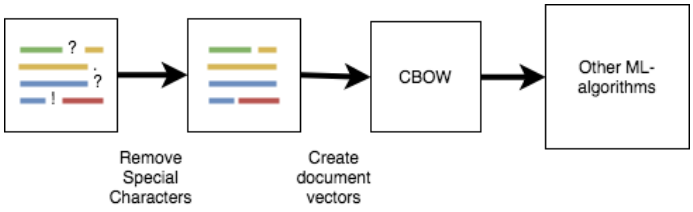

Fig. 2. The input vector of the algorithms: decision tree, random forest and multi layer perception is generated as shown above. It applies the filter layer conditions, and by using a CBOW algorithm directly on the document it produces the document vectors that can be categorized with the classification algorithms.

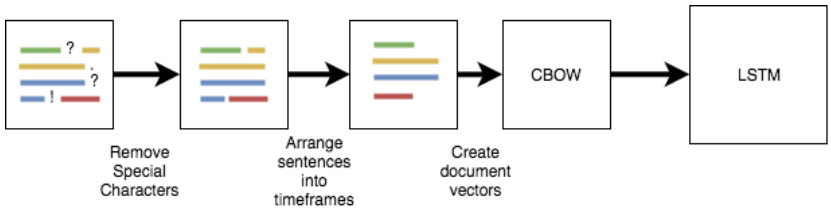

Fig. 3. The input vector of the LSTM algorithm is shown above. It first applies the filter layer conditions, then divides the document into sentences to be used as input data for the CBOW algorithm that produces the document vectors.

improved version [15], [16]. In the experiments in this article SyntaxNet is used with a Swedish training set (also called a treebank). Out of the resources mentioned in Nilson et al, we selected a treebank made by Jan Einarsson's project, which is well documented [17], [18], [19]. In the experiments we will try to select only nouns and verbs to predict a category.

Algorithm 1 A part-of-speech example sentence parsed by SyntaxNet.

Input sentence: I found a website to post AI tutorials .
Parsed dependency tree:

```
found VBD ROOT
+– I PRP nsubj
+– website NN dobj
|   +– a DT det
|   +– post VB infmod
|       +– to TO aux
|       +– tutorials NNS dobj
|           +– AI NNP nn
+– . . punct
```

C. Paragraph Vectors

The third step in Figure 1 is a neural network model that is constructed and trained to predict paragraph vectors when given the text form in the input or filter step. The paragraph vectors are unique vectors that describe the relation between the words in the document and a likely word to appear with them [2]. Computing the cosine similarity between two paragraph vectors yields a positive value when the documents are sharing similar contexts, a value close to zero when no relation could be found, and a negative number when a relation with opposite meaning [2]. With this knowledge, it is common to carry out paragraph operations such as you could for word vectors, for example Equation 2 [2].

$$king - man + woman = queen \tag{2}$$

The paragraph vectors do have context awareness, and are therefore believed to contain information about what makes a document category. The paragraph vectors are computed using the PV-DM algorithm which is an extension of the known word2vec algorithm bag of words (WV-BOW) [20].

The PV-DM algorithm tries to map all word vectors in a paragraph to a unique vector. The unique vector and the word vectors are averaged into the hidden layer h in our implementation. The rest of PV-DM algorithm follows the continuous bag of words (CBOW) algorithm [4], [21]. The unique paragraph vector can be considered an additional word in the context of a CBOW network. The idea of this extra vector is to have a form of memory about the topic of the paragraph, which explains the name PV-DM. The training of a PV-DM uses stochastic gradient descent [22] and neural network back-propagation by calculating the derivate of the vector from the next layer and applying it to compute the previous layer vector.

In our experiments, the PV-DBOW paragraph algorithm is implemented by the distributed memory vector concatenated with the distributed bag of words vector described by Mikolov et al [2].

D. Text Model

When a paragraph representation has been established it is time to go to step four: the categorization step. Therefore, we continue with the assumption that the paragraph vectors are properly and uniquely defined with good paragraph relationships in the previous step. The categorization algorithms we propose in these experiments are decision trees, random forest, multi layer perception and long-short term memory (LSTM).

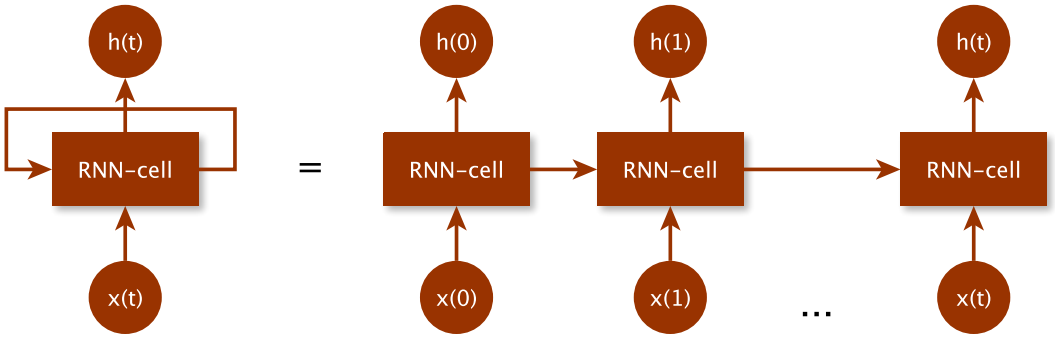

Fig. 4. A recurrent neural network cell. The cell to the left is the general notation of the hidden layer in an RNN model. The right unfolded version is the representative of the RNN with a short-term length of t.

Each algorithm has its own parameters that will be evaluated in the experiments. The input of the categorization step is the paragraph representation of the text. The output is the category belonging probabilities for each considered category described in Equation 1.

For the machine learning algorithms decision tree, random forest, multi layer perception are using the document vector from doc2vec CBOW algorithm was used directly as input, as shown in Figure 2. For the LSTM model, the time parameter was used to separate the paragraph into sentences, and then calling the CBOW algorithm in each sentence input into each LSTM time-slot as shown in Figure 3.

LSTM networks are based on recurrent neural network techniques. A recurrent neural network (RNN) is constructed like a neural network with an input, hidden layers and a output layer. The size of the input and output layer depends on the objective of training. The RNN-cell can be visualised as shown in Figure 4. The activation function of an RNN is usually the $tanh$ function. For each iteration, the model is trained by backpropagation through the network. The purpose of RNN is to have a short-term memory that remembers previous neurons. One of the first and simple constructions of RNN is the recurrent neural network language Model (RNN-LM). The hidden layer of an RNN-LM algorithm remembers the neurons one time-step back in the training history. [23]

Today, there are different variations of RNNs depending on the goal of the model. Andrej Karpathy summarises the different networks that are used into different mappings. One-to-one mapping is the original algorithm, for example the RNN-LM algorithm. One-to-many mapping when there is one input and several RNNs connecting to several outputs. This mapping can, for example be, used for image prediction with one image and several words that predict the image. Many-to-one mapping is where there are several inputs mapping to one output, this mapping can for example be used for classification. Many-to-many mapping is what Karpathy describes as two different mappings: one mapping that maps to an equal number of input and output RNNs (N-N), and another mapping that maps to a different number of inputs and outputs (N-M). The N-N mapping can, for example, be used to predict video sequences over time, while N-M mapping can be used for translation problems. [24]

LSTM networks are a special case of RNN that solve a fatal problem in the original RNN. The long-term problem that LSTM solves is introduced in RNN where the gradient descent exponentially diverges to infinity or converges to zero. On an ordinary RNN, the most simple solution that is used is to clamp the value between zero to one, but that still leaves the convergence to zero problem. The way LSTM networks work is to introduce three sigmoid layers and certain gates that only let parts of the information through to compensate for the vanishing gradient. The first sigmoid layer determines what information that is important from the previous LSTM-cell, the second sigmoid layer determines what information is important from the $tanh$ layer in the current cell and the third sigmoid layer determines what information will be passed to the next LSTM-cell. The gates that open or close based on the input from the previous LSTM-cell either remove or add information to a cell state that is also passed through to the next LSTM-cell. The third sigmoid layer extracts a piece of information from the cell state to the output value. [25], [26]

E. Output Category

As mentioned in the beginning of this chapter, the output layer interprets the output of the classification model and determines the number of categories, the probability of the prediction, and finally returns the category probabilities of the given text paragraph. The output category probabilities are normalized such that the highest probability of a category is one and the lowest one is zero according to Equation 3.

$$\frac{value - min(value)}{max(value) - min(value)} \tag{3}$$

IV. Evaluation and Experiments

This section presents the experiments conducted in the evaluation of the model. The dataset that we will train the models on consists of Swedish texts from the MittMedia article

TABLE I
THE PARAMETERS USED TO TRAIN THE CLASSIFIER MODELS. THERE ARE 2848 TRAINED MODELS IN TOTAL, E.G. EACH COMBINATION OF THE PARAMETERS FOR EACH ALGORITHM.

Algorithm	Parameter	Values
MLP	Dimensionality of the feature vectors	100x1, 100x2, 100x3, 100x4, 100x5 and 100x6
	Activation function	Identity, Logistic sigmoid, tanh and relu
	Solver function	LBFGS, SGD and Adam optimizer
	L2 penalty	0.005, 0.010, 0.015, 0.020
Decisiontree	Criterion	Gini and Entropy
	Max features	20%, 40%, 60%, 80% and 100% of the training data
	Max depth	10, 20, 30 and 40
	Minimum sample split	2, 4, 6 and 8
	Minimum leaf samples	2 and 4
Randomforest	Criterion	Gini and Entropy
	Max depth	10, 20, 30 and 40
	Minimum sample split	2, 4, 6 and 8
	Minimum leaf samples	2 and 4
	Number of trees	5, 10, 15, 20 and 25
	Features count for best split	2, 4, 8, 10, auto, $\sqrt{\text{number of features}}$, $\log_2$ (number of features)
LSTM	No of hidden layers	2, 3, 4
	LSTM neurons	10, 32, 50
	LSTM Timesteps	10, 20, 40
	Filter	Stop words, non-nouns and non-verbs
	Training epochs	2, 5, 15

database, including metadata. The metadata for categorization contains tags and categories that are attached to each training instance. Each instance can have more than one category. When the article was written, the categories were attached manually by the editors. During the experiments 5 to 30 categories were used to train, test and validate the models. The implementation of the experiments where made in Python. A tensorflow model was constructed for each model [27].

The data-instances are not always valid, therefore a prepossessing step is necessary to filter outlier texts. The dataset was divided into three groups to train, test and validate. First, one large filtered set was fetched from the database. The training and testing groups were separated into 60% training (5597 articles) and 40% testing (8396 articles) data after filtering of invalid outliers. Next a new non-seen filtered data-set was fetched and used for the validation group.

A. Experiment Settings

The following pre-processing was done before starting the actual training. The dataset used was filtered due to some odd outliers. The outliers are the result of different guidelines from the company Mittmedia, at different times, such as links to other articles or dynamically loading content. The restrictions were removed by ignoring the instance body content shorter than 10 words. If the instance body content contained more than 10 words, there were still outliers and unusable document-instances. The unusable instances sometimes contained JavaScript code that loaded contents from another URI onto the page dynamically when loading the page. Since these instances from the database are usually displayed in a web browser, this was not a problem. However, when the instances were directly fetched to the algorithm, the JavaScript content had to be removed.

In the experiments, the parameters of the classification algorithms consisted of all combinations of values for each algorithm as shown in Table I. The document count and categories were also changed independently of the algorithm parameters to evaluate impact on the result. For a full report on the results for the document and categories variation we recommend that you read the full report [20]. The F_1-score measurement were used to compare, validate and test the models. The measurement was developed in 1992 and gives an objective result of the harmonic mean between the precision and recall with equal weights [28].

The categories used for the experiment are labelled in Swedish: Blåljus, Ekonomi, Kultur, Nöje, Släkt och familj and Sport. The categories could be roughly translated as Accidents, Economy, Culture, Entertainment, Family and Sport, respectively. Accidents are texts about car chases, fires, injuries and so on. Economy is about financial issues such as business deals, the stock market and so on. Culture is mostly about art, museum or movie premiers, the nobelprice and so on. Entertainment is similar to Culture, as also this category potentially could include movie reviews, popular events, and other fun activities in the society. It is a fine line what would be defined as Culture and what is defined as Entertainment and different editors could have slightly overlapping definitions. Family is about newborns, the royal family, or family activities. The Sport category covers all kinds of sports, such as tennis, hockey, horse riding and so on. A majority of news

TABLE II
CONFUSION MATRIX OF THE CBOW AND LSTM COMBINED PREDICTIONS OF THE NEWS ARTICLE DATASET

Real \ Predicted	Accidents	Economy	Culture	Entertainment	Family	Sport	
Accidents	536	35	3	2	18	6	600
Economy	1	465	39	4	38	2	600
Culture	50	30	426	35	106	2	600
Entertainment	15	33	200	272	62	18	600
Family	14	25	112	29	413	7	600
Sport	14	17	10	25	81	453	600
	632	605	790	367	718	488	3600

TABLE III
CONFUSION MATRIX OF THE SINGLE LSTM-NETWORK PREDICTIONS OF THE NEWS ARTICLE DATASET

Real \ Predicted	Accidents	Economy	Culture	Entertainment	Family	Sport	
Accidents	287	51	16	27	46	173	600
Economy	154	287	27	34	57	41	600
Culture	32	100	262	13	121	72	600
Entertainment	92	128	106	139	58	77	600
Family	85	70	62	41	242	100	600
Sport	148	20	13	50	90	279	600
	798	656	486	304	614	742	3600

are written for the Sport category. Therefore it is important that we consider equal amount of text documents for each category so that the model isn't biased to, for example the Sport category.

TABLE IV
ACCURACY OF THE EVALUATED MODELS

Algorithm	Test Score	Validation Score
LSTM	0.74	0.71
LSTM (without CBOW)	0.42	0.37
MLP	0.31	0.14
Decision Tree	0.10	0.05
Random forest	0.08	0.03

The best model will be selected and evaluated without the CBOW vectors but, instead, word identifiers are used to verify that the combination is better than the algorithm alone. For objective fairness, the settings of the additional evaluated model will be the same as the best performing model.

B. Results

The test and validation measurements of each model are presented in Table IV. Only the best performing models are selected and presented in Table IV for each algorithm. The neural network is consistently performing well with a test F-score of about 0.31 and validation score of 0.22. The decision-tree classifier performs with a validation score of 0.05. The F-score of randomforest validation is 0.03. The LSTM network is currently superior to the other algorithms with a test score of 0.74 and validation score of 0.71. The confusion matrix of the combined CBOW and LSTM model show that the categories Culture and Entertainment are frequently mixed up by the algorithm, it is where the majority of miss-predictions occur, see Table II. In Table III the LSTM is compared with indicies as input, which shows that there is a larger uncertainty in this data.

Since the LSTM model performed best out of the the selected models, the additional model was trained using only the LSTM model with the same settings and unique IDs as input data. The additional model was performing with a validation score of 0.37 and thus we can confirm the research question that will investigate the combination of paragraph vectors and classification algorithm.

By running the model with different initial conditions we can evaluate the be model that had highest score value with a statistical approach. This way we check the reliability of the model in case it will be retrained at some point.

C. Discussion

The confusion matrix in Table III indicates that articles within one category are potentially difficult to distinguish from another category's texts. For example, the categories Entertainment, Family and Culture have some prediction overlaps. Most predictions are correct for all categories, which means that there are at least a few articles that characterize each category. Comparing LSTM with CBOW and the network without CBOW yields that the combination has significantly better performance. The reason is likely to have something to do with the vocabulary size, which is many times larger than the dataset that we are using, and thus not all words are present in the training data. This means that it is more difficult for LSTM without CBOW to predict correct categories.

Filtering away all words except nouns and verbs performed poorly compared to using all words in the document as input. The reason could be because the document vector also captures some information about how the text is structured and how the words are used in conjunction with each other. For example, which words tend to be used together, thus more frequent the word in a certain category the easier it is to predict. The time and network sizes did not significantly change the outcome, the score was slightly better using a larger value with any or both parameters. MLP is a good candidate if speed is a concern, although not near as good as fasttexts' performance. Random forest is slightly better than the decisiontree algorithm, but in general they perform similarly.

V. Conclusion

In this article we proposed a combination of classification algorithms and paragraph vector algorithms to improve the results of categorization problems. We aimed to find out if the combination of classification and paragraph vector algorithms improves the categorization, which we found to be true. We also investigated how the algorithm performed on news articles and to what extent it can be used. From the trained categorization models a probability score can be estimated for each available category that can be predicted. Based on the probabilities, a number of categories can be suggested to the editors in the system that, for example, has a probability higher than a certain threshold. The LSTM model is perfoming best in combination with the word vectors when predicting the categories. Based on the confusion matrix we can see that it is not overfitted. It can be concluded that a combination of LSTM and CBOW (classification and paragraph vector) algorithms perform better together (score of 0.71) than using only a classification algorithm such as LSTM (score of 0.37). Although the combination is not enough for the other evaluated algorithms: decisiontree, random forest and MLP with the combined CBOW algorithm achieve better result than a LSTM network with word IDs as input.

Future work for this project could be to extend the domain to other domains outside of the journalists that has a certain way of writing texts, such as common word choices and spelling standards. Although the proposed model should work in any other domain, further exploration has to be made to confirm. A recommended next step is to compare the model with the results of Liu et al, to make this possible we need to look into what data that model is evaluated on and see if we can apply the CBOW combined LSTM model using that data instead.

References

[1] O. Hjelmstedt and M. Sellfors, "Robotjournalistikens nya utmaningar," 2017.

[2] Q. V. Le and T. Mikolov, "Distributed Representations of Sentences and Documents." in *ICML*, vol. 14, 2014, pp. 1188–1196.

[3] A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov, "Bag of Tricks for Efficient Text Classification," *arXiv preprint arXiv:1607.01759*, 2016.

[4] S. Lai, L. Xu, K. Liu, and J. Zhao, "Recurrent Convolutional Neural Networks for Text Classification," *Twenty-Ninth AAAI Conference on Artificial Intelligence*, pp. 2267–2273, 2015.

[5] P. Liu, X. Qiu, X. Chen, S. Wu, and X. Huang, "Multi-timescale long short-term memory neural network for modelling sentences and documents," in *Proceedings of the 2015 conference on empirical methods in natural language processing*, 2015, pp. 2326–2335.

[6] A. Go, R. Bhayani, and L. Huang, "Twitter sentiment classification using distant supervision," *CS224N Project Report, Stanford*, vol. 1, no. 12, 2009.

[7] M.-L. Antonie and O. R. Zaiane, "Text document categorization by term association," in *Data Mining, 2002. ICDM 2003. Proceedings. 2002 IEEE International Conference on*. IEEE, 2002, pp. 19–26.

[8] M. Dabrowski, J. Gromada, and T. Michalik, "A practical study of neural network-based image classification model trained with transfer learning method." in *FedCSIS Position Papers, DOI: http://dx.doi.org/10.15439/2016F211*, 2016, pp. 49–56.

[9] A. Conneau, H. Schwenk, L. Barrault, and Y. Lecun, "Very deep convolutional networks for natural language processing," *arXiv preprint*, 2016.

[10] D. Połap, M. Woźniak, W. Wei, and R. Damaševičius, "Multi-threaded learning control mechanism for neural networks," *Future Generation Computer Systems, DOI: https://doi.org/10.1016/j.future.2018.04.050*, vol. 87, pp. 16–34, 2018.

[11] K. Leetaru and P. A. Schrodt, "Gdelt: Global data on events, location, and tone, 1979–2012," in *ISA Annual Convention*, vol. 2. Citeseer, 2013.

[12] M. D. Ward, A. Beger, J. Cutler, M. Dickenson, C. Dorff, and B. Radford, "Comparing GDELT and ICEWS event data," *Analysis*, vol. 21, pp. 267–297, 2013.

[13] A. Thanda, A. Agarwal, K. Singla, A. Prakash, and A. Gupta, "A Document Retrieval System for Math Queries," pp. 346–353, 2016.

[14] A. Voutilainen, "Part-of-speech tagging," *The Oxford handbook of computational linguistics*, pp. 219–232, 2003.

[15] D. Andor, C. Alberti, D. Weiss, A. Severyn, A. Presta, K. Ganchev, S. Petrov, and M. Collins, "Globally normalized transition-based neural networks," *arXiv preprint arXiv:1603.06042*, 2016.

[16] C. Alberti, D. Andor, I. Bogatyy, M. Collins, D. Gillick, L. Kong, T. Koo, J. Ma, M. Omernick, S. Petrov *et al.*, "Syntaxnet models for the conll 2017 shared task," *arXiv preprint arXiv:1703.04929*, 2017.

[17] J. Nilsson and J. Hall, *Reconstruction of the Swedish Treebank Talbanken.* Matematiska och systemtekniska institutionen, 2005.

[18] J. Einarsson, "Projektet talbanken. i: C platzack (utg), svenskans beskrivning 8, s76-96," 1974.

[19] ——, "Talbankens talspråkskonkordans," 1976.

[20] J. Lindén, "Understand and Utilise Unformatted Text Documents by Natural Language Processing algorithm," vol. 46, no. 0, 2017.

[21] X. Rong, "word2vec parameter learning explained," *CoRR*, vol. abs/1411.2738, 2014. [Online]. Available: http://arxiv.org/abs/1411.2738

[22] B. Recht, C. Re, S. Wright, and F. Niu, "Hogwild: A lock-free approach to parallelizing stochastic gradient descent," in *Advances in Neural Information Processing Systems*, 2011, pp. 693–701.

[23] T. Mikolov, S. Kombrink, L. Burget, J. Černocky, and S. Khudanpur, "Extensions of recurrent neural network language model," in *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*. IEEE, 2011, pp. 5528–5531.

[24] A. Karpathy, "The unreasonable effectiveness of recurrent neural networks," *Andrej Karpathy blog*, 2015.

[25] C. Olah, "Understanding lstm networks," *GITHUB blog, posted on August*, vol. 27, p. 2015, 2015.

[26] M. Sundermeyer, R. Schlüter, and H. Ney, "LSTM Neural Networks for Language Modeling." in *Interspeech*, 2012, pp. 194–197.

[27] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard *et al.*, "Tensorflow: a system for large-scale machine learning." in *OSDI*, vol. 16, 2016, pp. 265–283.

[28] N. Chinchor and B. Sundheim, "Muc-5 evaluation metrics," in *Proceedings of the 5th conference on Message understanding*. Association for Computational Linguistics, 1993, pp. 69–78.