Sequential Transfer Learning for Event Detection and Key Sentence Extraction

Anaïs Ollagnier
Department of Computer Science
University of Exeter
Exeter, United Kingdom
a.ollagnier@exeter.ac.uk

Hywel Williams
Department of Computer Science
University of Exeter
Exeter, United Kingdom
h.t.p.williams@exeter.ac.uk

***Abstract*—Sequential transfer learning (STL) techniques aim to improve learning in the target task by leveraging knowledge from a related domain using pre-trained representations. Approaches based on these techniques have achieved state-of-the-art results on a wide range of natural language processing (NLP) tasks. In the context of event detection and key sentence extraction, we propose to explore STL-based techniques using the last generation of pre-trained language representations, namely, ALBERT, BERT, DistilBERT, ELECTRA, OpenAI GPT2, RoBERTa and, XLNet. Experiments are conducted as a part of our contribution to the CLEF 2019 ProtestNews Track, which aims to classify and identify protest events in English-language news from India and China. Averaged results show that a STL-based method with OpenAI GPT2 outperforms prevailing methods in this domain by achieving better performance across event detection and key sentence extraction tasks. In addition, OpenAI GPT2 also obtains the best results on the majority of datasets tested in comparison to the best system presented during the CLEF 2019 ProtestNews challenge.**

***Index Terms*—transfer learning, pre-trained language models, neural networks, event detection**

I. Introduction

Traditional machine learning technology is based on the assumption that a difference in data distribution between the training data and test data may result in a degradation of the predictive learner [1]. Based on this paradigm, machine learning methods seek to make predictions of future outcomes using patterns learned from past information (training data) having the same input feature space and the same data distribution characteristics as testing data. Although these traditional methods have been successfully applied in many practical applications, it still has some limitations for certain real-world scenarios. Indeed, the success of real-world applications depends on the availability of sufficient labeled data to train machine learning models. However, collecting sufficient training data is often expensive and time-consuming, and may be unachievable in some scenarios. To overcome these limitations, *transfer learning* (TL) has been introduced with the aim of achieving high-performance learning outcomes by leveraging available data from different domains [2], [3]. TL uses prior knowledge gained from a source task to improve performance in a target task. Over the last few years, TF has achieved great success in many natural language processing (NLP) tasks, particularly when using *sequential transfer learning* (STL) methods [4]. The STL architecture, in which tasks are learned in sequence, has become a popular TL approach for NLP due to its ability to enable fast adaptation to a target task.

Recent advances in language pre-training have significantly improved upon the state-of-the-art on a wide range of NLP tasks, with prominent *pre-trained models* (PTM) such as BERT, RoBERTa, XLNet, ALBERT, and DistilBERT, amongst others [11]. The wide availability and ease of integration of these methods has led to the emergence of several STL architectures based on PTM [6], [4], [7]. It remains challenging to determine the best PTM model for a particular task. In this paper, we propose to evaluate several recent PTM combined with a STL architecture in the context of event extraction tasks. Experiments are conducted on datasets used in the CLEF ProtestNews challenge[1], which aims to extract event information from news articles across multiple countries. Using these public datasets, models are evaluated on two specific tasks: (Task 1) *news article classification* which consists of identifying news articles corresponding to political conflicts; (Task 2) *event sentence detection* which focuses on the identification of sentences referring to protest events. From these experiments, we identify PTM that achieve the best performance over each task individually and also both tasks combined. More broadly, this study evaluates the portability and generalisation ability of the PTM with regard to different data types (i.e. document-level in Task 1 and sentence-level in Task 2, articles from multiple countries in both tasks).

The contributions of this paper are: (1) Empirical comparison of STL techniques for event detection against state-of-the-art methods; (2) Evaluation of generalizability and reliability of STL techniques on both cross-context settings and different levels of scope.

II. Related work

In a STL scenario the source and target tasks are different and training is performed in sequence. Typically, STL consists of two stages: a *pre-training phase* in which general representations are learned on a source task or domain, and

[1] https://emw.ku.edu.tr/clef-protestnews-2019 Date of access: 7th September 2020.

an *adaptation phase* during which the learned knowledge is transferred to the target task or domain. Formally, as given in [4], the approach considers two tasks $\{T_1, T_2\}$ where T_1 runs over the interval $[i_1, i_2]$ and T_2 over the interval $[i_3, i_4]$. T_2 starts only once T_1 has terminated, which implies $i_2 \leq i_3$.

Similarly to other NLP tasks, various methods have been used to help extraction of suitable discriminant features at the pre-training phase. Unsupervised approaches (that learn patterns from unlabelled data) have become the most popular pre-training schemes, particularly those based on neural network approaches such as auto-encoding (data compression algorithm) and skip-thoughts (generic, distributed sentence encoder) models [8], [9], [10]. PTM shares the same idea of leveraging a large amount of unlabeled text to build a general model of language understanding, before being fine-tuned on specific NLP tasks [11].

Concerning event detection and key sentence extraction tasks, recent approaches have also been based on popular unsupervised models including *word2vec*, *GloVe* (global word-word co-occurrence statistic vectors), *FastText* (an extension of *word2vec* which includes character n-grams) and *ELMo* (state-of-the-art contextual word vectors) [12]. Recently a STL-based architecture combined with BERT (BidirectionalEncoder Representations from Transformers) has been introduced as part of an event extractor framework [13]. The CLEF Protest News Track, a competition introduced in 2019 to evaluate methods for event classification and detection, was won by a *Bidirectional Gated Recurrent Unit* (BiGRU) model using embedding vectors based on the Google News database [7]. During the CLEF Protest News challenge, most other proposed systems were derived from word embeddings (e.g. *GloVe*, *word2vec*), language modeling (e.g. *ELMo*) combined with neural networks (e.g. *Long Short-Term Memory* (LSTM), *Bidirectional LSTM* (BiLSTM), BiGRU) [6], [14], [15]. The main conclusions from the challenge were, firstly, that neural networks outperformed other models over each task, and secondly, that most submissions suffered from data dependence problems. These findings support our suggestion that STL-based architectures combined with PTM might permit models with high portability and better ability to generalize to new data.

III. System architectures

This section details the STL-based architecture used to conduct the experiments. There are two main stages: the *pre-training phase*, and the *adaptation phase*.

Pre-trained language representations are introduced during the pre-training phase. Table I provides a brief overview of each PTM[2] and describes the model architecture, the number of parameters and the training dataset used.

For the *adaptation phase*, empirical studies conducted on the development sets for each task[3] found best performance using a CNN LSTM classifier. Figure 1 details the architecture used and the shared parameters for both tasks. The model takes as input a time-ordered sequence of tokens (words) of arbitrary length (truncated to the average length of input sequences and then padded with zero vectors) and outputs a document-level or sentence-level prediction depending on the task. After the embedding layer, the layer corresponding to the CNN classifier (one-headed) is introduced using a configuration of 32 parallel feature maps and a kernel size of 3. Immediately afterwards, a LSTM layer is added (set to 100 internal units). Then, a dense layer of 64 nodes with ReLu is inserted. Finally, an output layer is used with one node containing sigmoid function for binary classification. The models have been trained using Adam as optimizer function set with a learning rate of $2e-5$ and batch size fixed to 16 for both tasks.

IV. Experimental Evaluation

A. Experimental Setup

The experiments were conducted using datasets provided at the CLEF ProtestNews Track 2019, which were developed to facilitate evaluation of text-based event detection systems. Here we focus on two sub-tasks related to this track, namely, news article classification (Task 1) and event sentence detection (Task 2). Task 1 requires identification of news articles associated with political conflicts through a binary classification scheme ('protest' vs. 'non-protest'). Task 2 focuses on identifying and labeling sentences that refer to protest events (e.g. riots, social events). The datasets[4] are composed of English-language news articles from India and China. They are provided with manual annotations that assign class labels. For both tasks, training and development datasets are extracted from a single source country (composed exclusively of news articles from India). Test datasets are provided from two countries; one test set is composed of news articles from India (here called the 'Source test' dataset) and another test set is from China (the 'Target test' dataset). Testing on data from two countries evaluates the portability/generalisability of the models. The test sets are not provided with labels and evaluation is conducted by submitting predicted labels to the Protest News team using an online platform (using the same URL as for dataset access). Table II summarises the distributions of each class in Tasks 1 and 2.

B. Experimental Results

Models were trained on a workstation with 36-core CPU and AMD FirePro W2100 GPU. Table III presents the results obtained in each task. Systems were evaluated according to the F1-scores for Task 1 and Task 2, and the average of F1-scores from both tasks (Avg.2).

1) Combined performance: According to the Avg.2 metric, OpenAI GPT2 produced the best model performance in comparison with the other STL-based models. RoBERTa is the second-ranked scheme with very similar performance

[2] All models are available online:
https://github.com/huggingface/transformers Date of access: 15th May 2020.

[3] Evaluations (not reported here) were conducted on LSTM, BiLSTM, BiGRU, CNN and CNN LSTM using the same architecture as presented here.

[4] Available online:
https://competitions.codalab.org/competitions/20288 Date of access: 14th May 2020.

TABLE I
DESCRIPTION OF PRE-TRAINED LANGUAGE MODELS.

Name	Architecture	Num. parameters	Trained data
ALBERT [16]	12 repeating layers, 128 embedding, 768-hidden, 12-heads	11M parameters	Trained on BookCorpus and Wikipedia. Training is performed on the base version.
BERT [17]	12-layer, 768-hidden, 12-heads	110M parameters	English texts from Wikipedia and Toronto BookCorpus. Original version trained on cased text is used here.
DistilBERT [18]	6-layer, 768-hidden, 12-heads	65M parameters	Based on English Wikipedia and Toronto Book Corpus. Base cased text version used here.
ELECTRA [19]	12-layer, 256-hidden, 12-heads	14M parameters	Uncased English text from Wikipedia and Toronto BookCorpus. Small discriminator version used here.
GPT-2 [20]	12-layer, 768-hidden, 12-heads	117M parameters	OpenAI's GPT-2 English model is based on 40GB of Internet text ($\simeq$ 8 million web pages). Here we used the small-sized version released by the company.
RoBERTa [21]	12-layer, 768-hidden, 12-heads	125M parameters	Based in five English-language corpora of varying sizes and domains (BooksCorpus, CC-News, OpenWebText and, STORIES). Training is performed on the base version trained on cased text.
XLNet [22]	12-layer, 768-hidden, 12-heads	110M parameters	Pre-trained models are based on English texts from Wikipedia, BooksCorpus, Giga5, ClueWeb, and Common Crawl. Here, the base version trained on cased text is used.

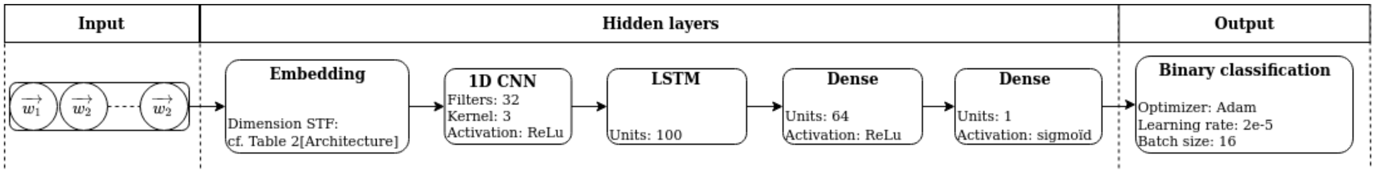

Fig. 1. Sequential model architecture details.

TABLE II
DESCRIPTION OF CLEF 2019 PROTESTNEWS DATASETS.

Task	Dataset	Protest	Not Protest	Total
Task 1	Train	769 (22.41%)	2,661 (77.58%)	3,430
	Dev.	102 (22.31%)	355 (77.68%)	457
	Source test	//	//	687
	Target test	//	//	2,303
Task 2	Train	988 (16.78%)	4,897 (83.21%)	5,885
	Dev.	138 (20.81%)	525 (79.18%)	663
	Source test	//	//	1,107
	Target test	//	//	1,235

(-0.004 Avg.2 relative to OpenAI GPT2). DistilBERT and ALBERT provide the least convincing results (-0.050 and -0.048 respectively). The best model run from the CLEF 2019 ProtestNews challenge outperforms OpenAI GPT2 by a small margin (-0.008), though the results vary between tasks. STL-based architectures combined with PTM achieved better performances than the best 2019 model for most datasets, with the exception of the Task 2 'Target' set.

2) Task 1 performance: For the 'source' dataset, where the model was tested on data from the same country as the training data, ELECTRA is the first-ranked scheme, slightly outperforming OpenAI GPT2 (-0.002 F1-score relative to ELECTRA) and showing good efficiency. ALBERT, which was one of the less efficient schemes, also performed well and was third-ranked (-0.010 from ELECTRA). For the 'target' set, BERT achieved the best F1-score narrowly followed by OpenAI GPT2 (-0.005 from BERT). Taking the mean F1-score over the source and the target sets, OPenAI GTP2 outperforms both the other STL-methods and the best model from CLEF 2019 with a mean F1-score of 0.748. Although the least convincing performances are obtained by DistilBERT (mean F1-score of 0.666), the largest performance gap between the two datasets (i.e. the worst generalisation) is obtained by XLNet. This suggests that, as a part of a STL scheme, the XLNet PTM is less suitable for data types which differ from the training data. Conversely, OpenAI GPT2 is the model with the highest ability to generalize on long sequences.

3) Task 2 performance: XLNet achieves the best F1-score on the 'source' dataset against the other STL-based models and the best model from CLEF 2019 (-0.050 from XLNet). OpenAI GPT2 is the second-ranked scheme (-0.022 from XLNet) narrowly followed by DistilBERT (-0.038 from XLNet). On the 'target' set, RoBERTa obtains the best performance of the STL-based models but is outperformed by the best model from CLEF 2019 ($+0.044$ from RoBERTa). ALBERT and DistilBERT achieved the lowest results with respectively -0.174 and -0.112 from RoBERTa. For Task 2, XLNet also obtains the best mean F1-score of the STL-based models with a mean F1-score of 0.574, while ALBERT obtains both the lowest mean F1-score (of 0.456) and the largest performance gap between the source and target datasets. BERT also performs poorly on these metrics. Regarding these findings, ALBERT is the less efficient on short sequences and less suitable confronted to new data. Conversely, XLNet is the model with the highest ability to generalize and make accurate predictions on short sequences.

TABLE III
EXPERIMENTAL RESULTS USING DATA FROM CLEF 2019 PROTESTNEWS TRACK.

Scheme	Task 1				Task 2				Avg. 2
	source	target	mean	time (secs)	source	target	mean	time (secs)	
Best_run_2019	0.807	0.597	0.702	//	0.631	0.553	0.592	//	0.647
Baseline	0.532	0.283	0.407	220.398	0.503	0.273	0.388	69.551	0.591
ALBERT	0.840	0.613	0.726	6041.655	0.593	0.319	0.456	1339.625	0.591
BERT	0.806	**0.653**	0.730	7176.175	0.584	0.418	0.501	2406.288	0.615
DistilBERT	0.778	0.554	0.666	4173.8	0.643	0.381	0.512	1578.285	0.589
ELECTRA	**0.850**	0.598	0.724	2498.027	0.591	0.462	0.527	919.092	0.625
OpenAI GPT2	0.848	0.648	**0.748**	7433.183	0.659	0.403	0.531	2901.705	**0.639**
RoBERTa	0.821	0.620	0.720	7458.571	0.609	**0.493**	0.551	2816.610	0.635
XLNet	0.812	0.542	0.677	9108.294	**0.681**	0.468	**0.574**	2722.150	0.625

4) Computational cost: Running-time cost is more important on Task 1 with an averaged computational cost of 6269.957 seconds. This finding is explained by the size of the inputs (314 in Task 1 and 24 in Task 2). For Task 1, XLNet incurs the highest running-time cost, while ELECTRA gets the best performances followed by DistilBERT. For Task 2, ELECTRA is still top-ranked, while GPT2 and RoBERTa are the slowest. Considering model characteristics (reported in Table I), the number of parameters to be learned increases the computational cost of training the model, but the complexity of the system is also a factor (e.g. ALBERT vs DistilBERT on Task 1). Overall, it is interesting to note that despite a much lower computational cost in comparison to the other STL-based models, ELECTRA provided good performance results for most datasets.

5) Overall evaluation: To sum up, OpenAI GPT2 shows the best portability by achieving the best performance on both short and long sequences and also the best ability to generalize over the testing sets on long sequences. XLNet generalizes better on short sequences. Considering computational cost, ELECTRA is the fastest model and achieves results that challenge the top-performing STL-based models for most datasets. To conclude, STL-based models using PTM during the pre-training stage show promising results in most cases and are likely to provide a strong basis for further development.

V. CONCLUSION

Pre-trained models (PTM) have become a popular method, achieving good results in a range of NLP tasks [11]. In this paper, we have evaluated several recent PTM methods in the context of sequential transfer learning (STL). In a STL scenario, where tasks are learned in sequence, prior knowledge gained from a source task are used to improve performance in a target task. Here experiments in the context of an event detection task have shown that OpenAI GPT2 achieves better performance on both short and long text sequences on the majority of datasets tested. ELECTRA also produces strong results, with a good balance between speed and performance. The main finding from this study is that STL allows improvements to generalizability and reliability of proposed models to handle heterogeneous data (English-language news articles from multiple countries) in comparison with prevailing methods in the domain.

In future works we plan to investigate existing techniques for addressing class imbalanced data (disparity between classes). Research in this area has shown promising results, with decreased rates of mis-classification [**?**]. Considering the dataset distribution in Table II, we believe that using techniques able to deal with imbalanced classes may help classifiers to reduce the over-classification of the majority group and so improve global classification performance.

VI. ACKNOWLEDGMENTS

This work was supported by the Institute of Coding which received funding from the Office for Students (OfS) in the United Kingdom.

REFERENCES

[1] H. Shimodaira, “Improving predictive inference under covariate shift by weighting the log-likelihood function,” Journal of statistical planning and inference, vol. 90, no 2, pp. 227–244, 2000.

[2] S. J.Pan, Q. Yang, “A survey on transfer learning”. IEEE Transactions on knowledge and data engineering, vol. 22, no 10, pp. 1345–1359, 2009.

[3] K. Weiss, T. M. Khoshgoftaar, D. Wang, “A survey of transfer learning”. Journal of Big data, vol. 3, no 1, pp. 9, 2016.

[4] S. Ruder, “Neural transfer learning for natural language processing”, Doctoral dissertation, NUI Galway, 2019.

[5] X. Qiu, T. Sun, Y. Xu, Y. Shao, N. Dai, X. Huang, “Pre-trained models for natural language processing: A survey”, arXiv preprint arXiv:2003.08271, 2020.

[6] A. Ollagnier, H. Williams, “Classification and Event Identification Using Word Embedding”, Working Notes of CLEF 2019 - Conference and Labs of the Evaluation Forum (CLEF), vol. 6, pp. 7, 2019.

[7] A. Safaya, “Event Sentence Detection Task Using Attention Model”, Working Notes of CLEF 2019 - Conference and Labs of the Evaluation Forum (CLEF). vol. 6, 2019.

[8] R. Kiros, Y. Zhu, R. R. Salakhutdinov, R. Zemel, R. Urtasun, A. Torralba, S. Fidler, “Skip-thought vectors”, Advances in neural information processing systems, pp. 3294–3302, 2015.

[9] F. Hill, K. Cho, A. Korhonen, “Learning distributed representations of sentences from unlabelled data”, arXiv preprint arXiv:1602.03483, 2016.

[10] L. Logeswaran, H. Lee, “An efficient framework for learning sentence representations”, arXiv preprint arXiv:1803.02893, 2018.

[11] X. Qiu, T. Sun, Y. Xu, Y. Shao, N. Dai, X. Huang, “Pre-trained Models for Natural Language Processing: A Survey”, arXiv preprint arXiv:2003.08271, 2020.

[12] W. Xiang, B. Wang, “A Survey of Event Extraction From Text”, IEEE Access, vol. 7, pp. 173111–173137, 2019.

[13] S. Yang, D. Feng, L. Qiao, Z. Kan, D. Li, “Exploring Pre-trained Language Models for Event Extraction and Generation”, Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, ACL, 2019.

[14] E. Maslennikova, “ELMo Word Representations For News Protection”, Working Notes of CLEF 2019 - Conference and Labs of the Evaluation Forum (CLEF), vol. 6, 2019.

[15] G. Skitalinskaya, J. Klaff, M. Spliethöver, "CLEF ProtestNews Lab 2019:Contextualized Word Embeddings for Event Sentence Detection and Event Extraction", Working Notes of CLEF 2019 - Conference and Labs of the Evaluation Forum (CLEF), vol.6, 2019.

[16] z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, R. Soricut, "ALBERT: A Lite BERT for Self-supervised Learning of Language Representations", arXiv preprint arXiv:1909.11942, 2019.

[17] J, Devlin, M.-W. Chang, K. Lee, K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding", arXiv preprint arXiv:1810.04805, 2018.

[18] V. Sanh, L. Debut, J. Chaumond, T. Wolf, "DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter", arXiv preprint arXiv:1910.01108, 2019.

[19] K. Clark, M. T. Luong, Q. V. Le, C. D. Manning, "Electra: Pre-training text encoders as discriminators rather than generators", arXiv preprint arXiv:2003.10555, 2020.

[20] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, "Language models are unsupervised multitask learners", OpenAI Blog, vol. 1, no 8, pp. 9, 2019.

[21] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, V. Stoyanov, "RoBERTa: A Robustly Optimized BERT Pretraining Approach", arXiv preprint arXiv:1907.11692, 2019.

[22] Z. Yang, Z. Dai, Y. Yang, J. G. Carbonell, R. Salakhutdinov, Q. V. L, "XLNet: Generalized Autoregressive Pretraining for Language Understanding", arXiv preprint arXiv:1906.08237, 2019.