Text summarization for Indian languages using pre-trained models

Aishwarya Krishnakumar, Fathima Naushin A R, Mrithula K L and B Bharathi

Department of CSE, Sri Siva Subramaniya Nadar College of Engineering,Tamil Nadu, India

Abstract

The notion of summarising is as old as the ancient Greek and Roman plays. The information available in various situations, from plays to meetings to non-fiction work, has been and is summarised for as long as we remember. This abstract itself is an example of the usage of the concept of summarization. In this era of technology, everything around us is digitized. People tend to develop ideas that perform activities that only humans were able to do before the innovation of modern technology. Summarizing text documents is one such example. Today, we have developed various NLP and AI models to perform text summarization. While efficient models exist for native English, little attention is given to Indian languages. This paper discusses the work done by SSNCSENLP in ILSUM Indian Language Summarization on the multilingual code-mixed text task of FIRE 2022. In this paper, we present a comparison of the performance of a few existing models. From our best-evaluated model, we were ranked among the top ten on the validation sets for all three Indian languages—English, Gujarati, and Hindi. To summarize the above mentioned languages we have used mT5_m2m_CrossSum, XL-Sum, Bert and the mT5-small models of which mT5_m2m_CrossSum generated precise summaries of the given text.

Keywords

generated summary, Indian languages, pre-trained model, mT5_m2m_CrossSum

1. Introduction

Natural Language Processing(NLP), the branch of computer science and artificial intelligence(AI), combines computational linguistics with statistical machine learning and deep learning models. It develops a rule-based model for human language, which provides computers the ability to process human language in the form of voice data or text. This enables the computers to read text, hear speech, and interpret it. NLP breaks down the language into tokens and tries to understand the relationship between the tokens. NLP offers several tasks which include sentimental analysis, word sense disambiguation, grammatical tagging, content categorization, text summarization, topic discovery and modeling,speech-to-text and vice-versa, and many more. These tasks face several challenges to be more accurate in what they do because human language is filled with ambiguities and, not to forget, the several languages in use or the usage of metaphors, sarcasm, idioms, and other grammatical usage exceptions. There are several NLP

Forum for Information Retrieval Evaluation, December 9-13, 2022, India

aishwarya2010328@ssn.edu.in (A. Krishnakumar); fathima2010192@ssn.edu.in (F. N. A. R); mrithula2010075@ssn.edu.in (M. K. L); bharathib@ssn.edu.in (B. B.)

https://www.ssn.edu.in/staff-members/dr-b-bharathi/ (B. B.)

0000-0001-7279-5357 (B. B.)

tools and approaches that resolves these challenges to its best. But these tools are very limited for low resource languages.

Text summarization uses NLP techniques to digest huge volumes of digital text in any form, such as, from articles or magazines or from social media, and create summaries and synopses for indexes, research databases, or busy readers who don't have time to read full text. It is the process of identifying most important meaningful information in a text document and compressing them into a shorter version. When the computer performs this task using algorithms and programs, it is called *Automatic Text Summarization.*

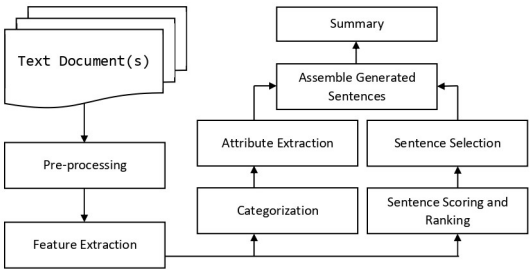

Figure 1: Basic flow of text summarizarion

There are two approaches to text summarization based on the *generated summary*(output):

- Extractive summarization.
- Abstractive summarization.

Extractive summarization uses simple and traditional algorithms to extract keyphrases from the text and use them to generate summaries. The keyphrases are usually chosen based on their frequency in the text. Thus, the generated summaries can be grammatically strange. Abstractive summarization uses advanced algorithms to create new phrases and sentences and generate summaries to convey the most useful information from the text. While the abstractive approach overcomes the grammatical inconsistencies in the extractive approach and performs better, it is also more difficult to develop the algorithms for the abstractive approach. The content of the generated summary can either be indicative or informative. An indicative summary represents only the main idea of the text document, whereas an informative summary gives a brief description of the text document. Indicative summaries are shorter than informative summaries. There are several approaches within these summarization approaches [1]. There are two types of summarization based on the *text document*(input):

- Single document summarization.
- Multi-document summarization.

Single document summarization generates summary of text from a single document while multi-document summarization generates summary from more than one document. Multi-document summarization is more difficult than single document because of the redundancy of text, compression of text from multiple documents, collection of significant information etc. MEAD, is a multi-document summarizer proposed in the paper [2]. There are broadly three categories of summarization based on the *purpose* of the generated summary:

- Generic summarization.
- Domain-specific summarization.
- Query-based summarization.

Generic Summarization condenses the overall information content available in the source text. Domain-specific summarization generates summaries from documents or text related to the given domain. Query-based summarization generates summaries that answer the search query. The query is given as an input along with the text document.Monolingual summarization generates summaries for a particular language domain, whereas multilingual summarization generates summaries for more than one language. While large-scale datasets exist for a number of languages like English, Chinese, French, German, Spanish, etc., no such datasets exist for any Indian languages. In this paper, we will use a multilingual pre-trained model to summarise text in Indian English and two other Indian languages—Hindi and Gujarati.

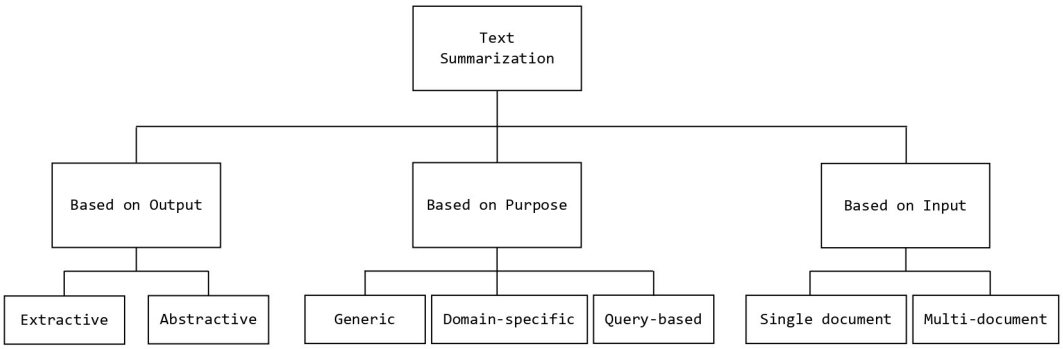

Figure 2: Categories in text summarizarion

The paper is organised as follows: The literature works related to text summarization are given in Section 2. The dataset analysis and task descriptions are given in Section 3. Section 4 details the different models experimented during the course of the shared task. Section 5 describes the internal architecture and working of the used pre-trained model mT5_m2m_CrossSum 4.1. Section 6 provides the performance metrics used to evaluate a summarization task. Section 7 provides the performance results of the shared task for the model submitted. Finally, Section 8 concludes the paper.

2. Related works

The majority of works are on extractive summarization because it is simple to implement.The complexity of natural language processing makes abstractive summarization a challenging task. Research on extractive summarization has plateaued after reaching its peak performance. Now the focus of researchers has shifted to abstractive summarization and the fusion of extractive and abstractive techniques. Abstractive summarization helps resolve the dangling anaphora problem and thus helps generate readable, concise, and cohesive summaries.

The author of the paper [3] focuses on approaches for text summarization that initially use extractive summarization techniques, followed by abstractive summarization techniques. The paper provides a brief explanation of how combining various extractive summarization techniques works. It demonstrates that fusion systems can assist us in improving the consistency of the meta-system. The author discusses an alternative technique to abstractive summarization, known as the *Generative approach* for text summarization. Their experiments also include changing the informativeness criteria used in abstractive summarization from TextRank scores of words to Log-Likelihood ratios of the words. The paper also proposes an approach that uses statistical machine translation for document summarizations.

Jagadish S Kallimani, et al in his paper, A Comprehensive Analysis of Guided Abstractive Text Summarization, suggests a solution for abstractive summarization of native Indian languages [4].In this method the abstract data is extracted and processed to gather the key concepts from the original text using extraxtive summarization techniques. Earlier research on summarizing documents in Indian languages adopted paradigms for extracting salient sentences from text using features like word frequency and phrase frequency, position in the text and key phrases. Such extractive summaries tend to have long sentences and the desired information is scattered across the document.

3. Dataset analysis and task description

The primary goal of this shared task is to generate a meaningful fixed-length summary, either extractive or abstractive, for the dataset's code-mixed and script-mixed articles in Dravidian languages (Hindi-English and Gujarati-English) and English. The news articles contain more than one sentence, and the heading and link of each article are given. Each article may contain English phrases even if the article itself is written in an Indian language. The task is to summarise the news article at an appropriate length.The detailed descripiton of the task, Findings of the First Shared Task on Indian Language Summarization (ILSUM): Approaches, Challenges and the Path Ahead are present in [5] [6].

4. Methodologies

This paper reflects on a specific approach of abstractive text summarization applied to English and Indian languages like Hindi and Gujarati. In terms of model architecture, we focus on approaches based on now-ubiquitous large-scale pre-trained language models (LM), such as XL-Sum, cross sum, BERT (Devlin et al., 2019) and BERT (Lewis et al., 2020), which obtained new state-of-the-art results in diverse natural language processing tasks, including text summarization [7] [8] [9]

4.1. mT5_m2m_CrossSum

We have used mT5_m2m_CrossSum, a large-scale cross-lingual abstractive summarization that has both the properties of the basic mt5 model and the fine-tuned m2m model. The LaSE, a

new metric for automatically evaluating model-generated summaries and showing a strong correlation with ROUGE is used to analyse the performance of the model in addition to usually existing measures like rouge and F1 scores. Performance on ROUGE and LaSE indicate that pre-trained models fine-tuned on CrossSum consistently outperform baseline models, even when the source and target language pairs are linguistically distant. CrossSum is the largest cross-lingual summarization dataset and the first-ever that does not rely solely on English as the pivot language.This model was the best to summarize the Hindi and Gujarati datasets [10]. The sample summaries generated by this model for English, Hindi and Gujarati datasets are presented in 3.

LANGUAGE	ARTICLE LINK	GENERATED SUMMARY
ENGLISH	https://www.indiatvnews.com/news/world/more-contagious-virus-variant-found-long-island-new-york-united-states-677384	The US state of New York has confirmed a new variant of coronavirus.
GUJARATI	https://www.divyabhaskar.co.in/local/gujarat/surendranagar/news/in-morbi-irregularities-in-impact-fee-collection-came-to-light-with-three-temporary-employees-fired-129750001.html	મોરબી નગરપાલિકાના નવનિયુક્ત ચીફ ઓફિસર દ્વારા ગેરકાયદેસર બાંધકામો છડેચોક બંધાઈ રહ્યા છે.
HINDI	https://www.indiatv.in/india/national/india-successfully-test-fires-the-new-generation-agni-prime-missile-827709	भारत ने ओडिशा तट पर 'अग्नि प्राइम' मिसाइल का सफल परीक्षण किया है.

Figure 3: A Sample of summaries generated by mT5_m2m_CrossSum model in all three languages

4.2. XL-Sum

Contemporary works on abstractive text summarization have focused primarily on high-resource languages like English, mostly due to the limited availability of datasets for low/mid-resource ones. In this work, we have used XL-Sum, a highly abstractive, concise, and high quality pre-trained model, as indicated by human and intrinsic evaluation.XL-Sum induces competitive results compared to the ones obtained using similar models that work only with monolingual datasets.This model gives a relatively high rouge score when compared to other models that follow. XL-Sum is the largest abstractive summarization dataset in terms of the number of samples collected from a single source and the number of languages covered.XL-Sum provided the highest performance scores for the English dataset [11].

	Rouge 1	Rouge-2	Rouge-3
Precision	0.1892	0.0637	0.0357
Recall	0.0919	0.0317	0.0178
F1-Score	0.1185	0.0407	0.0223

Table 1
Scores for generating summaries using XL-Sum for the English validation set.

4.3. BERT

We use a new language representation model called BERT, which stands for *Bidirectional Encoder Representations from Transformers*. Unlike recent language representation models, BERT is designed to pre-train deep bidirectional representations from unlabeled text by jointly conditioning on both left and right context in all layers. As a result, the pre-trained BERT model can be fine-tuned with just one additional output layer to create state-of-the-art models for a wide range of tasks, such as question answering and language inference, without substantial task-specific architecture modifications.BERT is conceptually simple and empirically powerful. It obtains new state-of-the-art results on eleven natural language processing tasks, including pushing the GLUE score to 80.5% (7.7% point absolute improvement), MultiNLI accuracy to 86.7% (4.6% absolute improvement), SQuAD v1.1 question answering Test F1 to 93.2 (1.5 point absolute improvement) and SQuAD v2.0 Test F1 to 83.1 (5.1 point absolute improvement). The BERT model cannot be used for any other languages but for English [12].

	Rouge 1	**Rouge-2**	**Rouge-3**
Precision	0.0917	0.0781	0.0236
Recall	0.0829	0.0213	0.0379
F1-Score	0.0861	0.3347	0.2098

Table 2
Scores for generating summaries using Bert for the English validation set.

4.4. mT5-small

With mT5-small, a multilingual variant of T5, reframing all NLP tasks into a unified text-to-text-format where the input and output are always text strings, in contrast to BERT-style models that can only output either a class label or a span of the input can be done. The text-to-text framework uses the same model, loss function, and hyperparameters on any NLP task, including machine translation, document summarization, question answering, and classification tasks (e.g., sentiment analysis). The mt5-small model can be used to train and validate a small datset and is relatively slow for training huge datasets when compared to all the models [13].

	Rouge 1	**Rouge-2**	**Rouge-3**
Precision	0.0707	0.0413	0.0119
Recall	0.0917	0.0109	0.0246
F1-Score	0.0798	0.0172	0.0163

Table 3
Scores for generating summaries using mT5-small for the Gujarati validation set.

All the models described above were used to generate the summaries for the given validation and test datasets.

5. Architecture

The internal architecture and working of the used pre-trained mT5_m2m_CrossSum is given below. This section explains a method which combines information extraction with summarization to produce a guided summary of domain specific documents. This method uses a narrower view which is to identify instances of a particular class of events and extract arguments relevant to this class of events.A fully abstractive approach with a separate process for the analysis of the text, the content selection, and the generation of the summary has the potential for generating summaries at a level comparable to humans. The presented method uses a rule-based, custom-designed IE module, along with categorization, content selection and sentence generation systems to fulfil the needs of abstractive summarization. The system uses repositories like rules and gazetteers to refer to the language syntax and semantics.This novel IE rule-based approach attempts to extract relevant information using lexical analysis tools like Part of Speech Tagging (POST) and Named Entity Recognition (NER). This ensures an information rich summary that reduces redundancy in not just the sentences produced but also in the information conveyed. Figure 4 shows the diagrammatic representation of the architecture for an abstractive summary generation system [14] [15] [16].

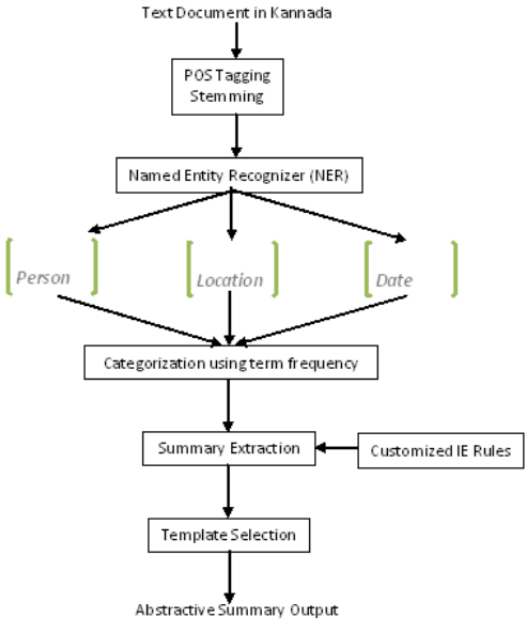

Figure 4: Proposed Architecture for Abstractive Summary Generation System

6. Performance analysis

The generated summaries for each language are evaluated using standard ROUGE metrics: Rouge-1, Rouge-2, and Rouge-4 F-scores. ROUGE stands for *Recall-Oriented Understudy for*

Gisting Evaluation. There are four different ROUGE measures as describe in [17]. The scores are computed by comparing the set of generated summaries against a set of reference summaries (typically human-produced). ROUGE-1 refers to the overlap of unigrams between the generated summaries and reference summaries; ROUGE-2 refers to the overlap of bigrams between the generated summaries and reference summaries; and so on. Precision, Recall and F1-score are the F scores.

Recall (in the context of ROUGE) refers to how much of the reference summary the generated summary captures, i.e., it is the fraction of sentences chosen by the human that were also correctly identified by the system.

$$Recall = \frac{|\text{overlap of generated and reference summary}|}{|\text{reference summary}|} \tag{1}$$

Precision refers to how much of the generated summary was in fact relevant or needed, i.e., it is the fraction of system sentences that were correct.

$$Precision = \frac{|\text{overlap of generated and reference summary}|}{|\text{generated summary}|} \tag{2}$$

If we just consider individual words, |generated summary| and |reference summary| refers to the number of words in the generated summary and reference summary respectively, whereas the |overlap of generated and reference summary| refers to the number of words overlapped words between the reference summary and the generated summary.

F1-score, also known as the F-measure or the F-score conveys the balance between the precision and the recall.

$$F1 - score = \frac{2 * Precision * Recall}{Precision + Recall} \tag{3}$$

These performance metrics are directly proportional to the number of overlapping words between the generated summary and the reference summary. It does not take into account the type of summary generated. As we saw in Section 1, the summary generated can be extractive or abstractive. Because extractive summarization uses the methodology of extracting key words from the given summary, it produces a higher number of overlapping words, even if they lack meaning. Whereas an abstractive summarization does not extract keyphrases, it generates a meaningful summary which conveys the text document at its best. This might not lead to a greater number of overlapping words,thereby falling short in the final scores calculation in terms of efficiency [18].

The summaries generated by abstraction are more meaningful and give the perfect gist of the contents to be summarized. Out of all the models used to generate summaries, the mT5_m2m_CrossSum model has provided accurate summaries for the given news article dataset across Indian languages.

7. Result

The mT5_m2m_CrossSum model described in Section 4.1 was used to submit the validation datasets for all three languages—English, Gujarati, and Hindi. Our submission secured the 7th

rank in the task on the Hindi dataset. The final performance results for the validation dataset in Hindi are recorded in Table 4.

	Rouge-1	Rouge-2	Rouge-3	Rouge-4
F1-Score	0.371	0.223	0.168	0.132
Precision	0.535	0.321	0.241	0.191
Recall	0.299	0.181	0.137	0.108

Table 4
Performance of the mT5_m2m_CrossSum model with a validation dataset for Hindi

Our submission secured the 6th rank in the task on the Gujarati dataset. The final performance results for the validation dataset in Gujarati are recorded in Table 5.

	Rouge-1	Rouge-2	Rouge-3	Rouge-4
F1-Score	0.119	0.040	0.022	0.014
Precision	0.189	0.063	0.035	0.021
Recall	0.091	0.031	0.018	0.011

Table 5
Performance of the mT5_m2m_CrossSum model with a validation dataset for Gujarati

Our submission secured the 9th rank in the task on the English dataset. The final performance results for the validation dataset in English are recorded in Table 6.

	Rouge-1	Rouge-2	Rouge-3	Rouge-4
F1-Score	0.274	0.089	0.044	0.025
Precision	0.428	0.144	0.073	0.043
Recall	0.210	0.067	0.033	0.019

Table 6
Performance of the mT5_m2m_CrossSum model with a validation dataset for English

8. Conclusion

In this paper, we have briefly described about the existing text summarization methods for Indian languages. We have showed the results and performance analysis of a few techniques. We have worked using mT5_m2m_CrossSum, XL-Sum, Bert and the mT5-small models of which mT5_m2m_CrossSum gave us the best results. Though models like Bert could be used only for English datasets, multilingual cross models outperformed the pre-trained monolingual models. We hope that this paper will gave a fair idea on the different models that can be used to summarize Indian English and Indian languages effectively on a given dataset of news articles collected overtime.

	Rouge-1	Rouge-2	Rouge-3	Rouge-4
F1-Score	0.37	0.23	0.17	0.14
Precision	0.55	0.33	0.25	0.20
Recall	0.30	0.18	0.14	0.11

Table 7
Performance of the mT5_m2m_CrossSum model with test dataset for Hindi

References

[1] N. Andhale, L. Bewoor, An overview of text summarization techniques, in: 2016 International Conference on Computing Communication Control and automation (ICCUBEA), 2016, pp. 1–7. doi:10.1109/ICCUBEA.2016.7860024.

[2] D. R. Radev, H. Jing, M. Styś, D. Tam, Centroid-based summarization of multiple documents, Information Processing Management 40 (2004) 919–938. URL: https://www.sciencedirect.com/science/article/pii/S0306457303000955. doi:https://doi.org/10.1016/j.ipm.2003.10.006.

[3] P. Mehta, From extractive to abstractive summarization: A journey, 2016, pp. 100–106. doi:10.18653/v1/P16-3015.

[4] J. S. Kallimani, K. Srinivasa, B. E. Reddy, A comprehensive analysis of guided abstractive text summarization, International Journal of Computer Science Issues (IJCSI) 11 (2014) 115.

[5] S. Satapara, B. Modha, S. Modha, P. Mehta, Findings of the first shared task on indian language summarization (ilsum): Approaches, challenges and the path ahead, in: Working Notes of FIRE 2022 - Forum for Information Retrieval Evaluation, Kolkata, India, December 9-13, 2022, CEUR Workshop Proceedings, CEUR-WS.org, 2022.

[6] S. Satapara, B. Modha, S. Modha, P. Mehta, Fire 2022 ilsum track: Indian language summarization, in: Proceedings of the 14th Forum for Information Retrieval Evaluation, ACM, 2022.

[7] D. Radev, S. Teufel, H. Saggion, W. Lam, J. Blitzer, A. Celebi, H. Qi, E. Drabek, D. Liu, Evaluation of text summarization in a cross-lingual information retrieval framework, Center for Language and Speech Processing, Johns Hopkins University, Baltimore, MD, Tech. Rep 6 (2002).

[8] Y. Liu, M. Lapata, Text summarization with pretrained encoders, arXiv preprint arXiv:1908.08345 (2019).

[9] K. Hong, A. Nenkova, Improving the estimation of word importance for news multi-document summarization, in: Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, 2014, pp. 712–721.

[10] A. Bhattacharjee, T. Hasan, W. U. Ahmad, Y.-F. Li, Y.-B. Kang, R. Shahriyar, Crosssum: Beyond english-centric cross-lingual abstractive text summarization for 1500+ language pairs, 2021. URL: https://arxiv.org/abs/2112.08804. doi:10.48550/ARXIV.2112.08804.

[11] T. Hasan, A. Bhattacharjee, M. S. Islam, K. Samin, Y.-F. Li, Y.-B. Kang, M. S. Rahman, R. Shahriyar, Xl-sum: Large-scale multilingual abstractive summarization for 44 languages,

arXiv preprint arXiv:2106.13822 (2021).
[12] Y. Liu, Fine-tune bert for extractive summarization, arXiv preprint arXiv:1903.10318 (2019).
[13] L. Xue, N. Constant, A. Roberts, M. Kale, R. Al-Rfou, A. Siddhant, A. Barua, C. Raffel, mt5: A massively multilingual pre-trained text-to-text transformer, arXiv preprint arXiv:2010.11934 (2020).
[14] G. Shilpa, D. Shashi Kumar, Abs-sum-kan an abstractive text summarization technique for an india regional language by induction of tagging rules, Int J Recent Technol Eng (IJRTE), ISSN (2019) 2277–3878.
[15] K. Cho, B. van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, Y. Bengio, Learning phrase representations using RNN encoder–decoder for statistical machine translation, in: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Association for Computational Linguistics, Doha, Qatar, 2014, pp. 1724–1734. URL: https://aclanthology.org/D14-1179. doi:10.3115/v1/D14-1179.
[16] T. Cohn, M. Lapata, Sentence compression beyond word deletion, in: Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008), Coling 2008 Organizing Committee, Manchester, UK, 2008, pp. 137–144. URL: https://aclanthology.org/C08-1018.
[17] C.-Y. Lin, ROUGE: A package for automatic evaluation of summaries, in: Text Summarization Branches Out, Association for Computational Linguistics, Barcelona, Spain, 2004, pp. 74–81. URL: https://aclanthology.org/W04-1013.
[18] J. Howard, S. Ruder, Universal language model fine-tuning for text classification, in: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Melbourne, Australia, 2018, pp. 328–339. URL: https://aclanthology.org/P18-1031. doi:10.18653/v1/P18-1031.