CROSS-LINGUAL DOCUMENT CLUSTERING FOR SINHALA, TAMIL, AND ENGLISH USING PRE-TRAINED MULTILINGUAL LANGUAGE MODELS

Malavarayar Vijayanandan Vithulan

(209390R)

Degree of Master of Science in Computer Science

Department of Computer Science and Engineering

University of Moratuwa
Sri Lanka

July 2022

CROSS-LINGUAL DOCUMENT CLUSTERING FOR SINHALA, TAMIL, AND ENGLISH USING PRE-TRAINED MULTILINGUAL LANGUAGE MODELS

Malavarayar Vijayanandan Vithulan

(209390R)

Thesis report submitted in partial fulfilment of the requirements for the degree Master of Science in Computer Science specialisation in Data Science.

Department of Computer Science and Engineering

University of Moratuwa
Sri Lanka

July 2022

DECLARATION

I declare that this is my work. This dissertation does not incorporate without acknowledgment any material previously submitted for a Degree or Diploma in any other University or institute of higher learning. To the best of my knowledge and belief, it does not contain any previously published material written by another person except where the acknowledgment is made in the text.

Also, I hereby grant the University of Moratuwa the non-exclusive right to reproduce and distribute my thesis/dissertation, in whole or in print, electronic, or another medium. I retain the right to use this content in whole or part in future works (such as articles or books).

Signature: Date: 05/05/2022

The above candidate has researched the Master's thesis Dissertation under my supervision.

Name of the supervisor: Dr. Surangika Ranathunga

Signature of the supervisor: Date:

ABSTRACT

Organising text articles into groups or clusters is known as document clustering. Documents that belong to a cluster are about the same subject. Document embeddings should be in the same embedding space for the cross-lingual document clustering, i.e., similar documents should have similar vectors. Obtaining document embedding for Tamil and Sinhala is feasible using models like Word2Vec or FastText, however, these embeddings are language specific, i.e., these will not be in the same vector space. Therefore, one cannot cluster documents across the languages using the language specific models. Pre-trained multilingual language models such as mBERT, XLM-R were introduced to solve this problem by transferring the knowledge from high resource languages to low resource languages.

This research is conducted to cluster Tamil, Sinhala and English news articles using XLM-R models. An adequate amount of collected documents were clustered, and the clustering techniques and performance were evaluated. This research produces a new baseline for cross-lingual clustering of Tamil, Sinhala, and English documents.

Keywords: Cross-lingual document clustering, Multilingual language models, XLM-R, mBERT, LASER, Knowledge distillation

ACKNOWLEDGEMENT

I want to express my deep and sincere gratitude to Dr. Surangika Ranathunga for guiding me in finding an exciting research topic and for continued support and encouragement. Her supervision immensely helped me in setting goals and engaging in the study.

I want to express my most incredible gratitude to the Department of Computer Science and Engineering, the University of Moratuwa, for providing the support to overcome this effort. Last but not least, my heartfelt gratitude goes to my parents and friends who supported me throughout this endeavour.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

Abbreviation	Description
NLP	Natural Language Processing
NER	Named-entity recognition
MLLM	Multilingual Language Models
mBERT	Multilingual BERT

1.0 INTRODUCTION

The process of organising text articles into groups or clusters is known as document clustering [1]. Documents that belong to a cluster are about the same subject, whereas documents belonging to another cluster are about different subjects. This has to be done automatically without manual intervention. It is widely employed in information retrieval, pattern recognition, knowledge discovery and data mining [1].

State-of-the-art performance in monolingual document clustering in English and other languages was achieved using TF-IDF, Doc2vec [29, 30, 4], FastText [3] and BERT [30] over the years. The advent of Transformer-based architecture and Transformer-based pre-trained models has revolutionised the field of Natural Language Processing (NLP) and has led to state-of-the-art performance on a wide range of tasks. Although there has been a surge of interest in general-purpose sentence representation, research in that area has been essentially monolingual and primarily focused on English. Cosine similarity measure was primarily used in most word embedding-based document clustering approaches [3, 30]. Several clustering techniques such as Agglomerative and K-Means have been used. Surprisingly, the one-pass clustering algorithm has provided better results than traditional clustering approaches in few studies [3, 4].

Recent advances in NLP have led to multilingual language models (MLLM) such as multilingual BERT (mBERT) with 104 languages from Google [29] and XLM-R with 100 languages from Facebook [2]. Low resource languages such as Tamil and Sinhala can benefit from these pre-trained MLLMs. M-BERT has only Tamil from Sri Lankan local languages and XLM-R has both Tamil and Sinhala in it. Cross-lingual document clustering can be achieved by employing one of these MLLMs.

However, good performance from these MLLMs cannot be achieved out of the box for low-resource languages primarily due to the following reasons.

1. Pre-trained models are trained generically, So in order to perform better in specific tasks like Named-entity recognition (NER), sentiment analysis, question answering, etc, task-specific fine-tuning is required.
2. Amount of low resource language data in the pre-trained model is comparatively less. Thus, cross-lingual vector alignment needs to be done.

To the best of our knowledge, there were no studies published on cross-lingual document clustering using pre-trained MLLMs. This research will be one of the initial studies in this domain, especially for Sri Lankan local languages - Tamil, Sinhala and English.

1.1 Research problem

Previously Tamil news clustering was done using monolingual embedding - fastText [3] and Sinhala news clustering with frequency vectors [4]. However, these models are monolingual, and each time it has to be trained for one language. Further, there are not any implementations yet to look at similar content articles from all three local languages at a glance.

1.2 Research objectives

The objective of this research is to implement a mechanism to form cross-lingual clusters using pre-trained MLLM for documents that are written in English, Tamil and Sinhala languages.

2.0 LITERATURE REVIEW

A comprehensive literature review in text representation, text clustering and multilingual language modelling is conducted in this chapter.

2.1 Text representation

Text-based data growth is tremendous due to sources such as social media, online forums and published articles. Text data is rich in information that contains more valuable insights than quantitative data. The primary goal of NLP is to understand the text in a human-like nature. Text representation has long been a focus of research in the field of NLP. Researchers have discovered an adequate amount of Machine learning algorithms with high accuracy. If the text can be quantitatively represented, NLP also can benefit from the existing Machine learning algorithms. A significant challenge in text representation is to keep the syntactic and semantic relationship between the words even after the representation. Text has been used in a variety of applications over the years, including document clustering, email filtering, sarcasm detection, sentiment and opinion mining, question answering, content mining, and many others [19]. This subsection is devoted to thoroughly researching the evolution of text representation methodologies from their inception to the current state-of-the-art models.

2.1.1 Classical methods

Early-stage text representation models were built on the frequency of words.

2.1.1.1 One hot encoding

This is the most basic way of representing text. The vector's dimension corresponds to the size of the vocabulary in the text/document. Each term in the dictionary is represented by a binary variable, such as 1 or 0. Index of the corresponding term will be 1 and everything else will be 0. [19]

2.1.1.2 Bag-of-Words (BoW)

The BoW encoding is simply an extension of the one hot encoding. It encapsulates the one-hot encoding of terms in a sentence. As vocabulary grows, so does the dimension of the BoW vector. Furthermore, a large number of zeros will introduce sparse matrix problems. [19]

2.1.1.3 Term Frequency (TF)

Term Frequency is the number of occurrences of a word calculated and divided by the number of words in a text. Therefore, most occurred terms will have a higher TF value compared to least occurred terms [19].

2.1.1.3 Term Frequency-Inverse Document Frequency (TF-IDF)

TF has the limitation of being biassed to most occurring terms such as stop words. TF-IDF eliminates the limitation of the Karen [19] proposed TF-IDF in 1988. TF stands for the Term Frequency as mentioned in 2.1.1.3. IDF stands for inverse document frequency and plays a vital role in reducing the impact of most occurring words such as stop words by providing higher weight to least occurring terms and lower weight to most occurring terms. Fayaza and Ranathunga [3] and Nanayakkara and Ranathunga [4] have used TF-IDF to set the benchmark for text representation for the news clustering or Tamil and Sinhala (respectively).

The BoW is used to build the TF-IDF. As a result, it is unable to retain word order or syntactic and semantic information. The TF-IDF is mathematically represented below [6, eq. (1)].

$$TF - IDF(t, d, D) = TF(t, d) \times log(\frac{D}{dft}) \qquad \text{[eq. (1)]}$$

Here, *t* represents the terms, and *d* represents each document. *D* denotes a collection of documents, and dft denotes the sum of documents containing the term *t* [19].

These models, however, have a high dimensionality and do not capture the syntactic and semantic relationships between words. Due to these constraints, research on decentralised language models in low dimensional space has been conducted.

2.1.2 Word Embeddings

Word embedding is a method in NLP that represents the words as vectors. Similar words would have a similar vector (The distance between both vectors is small). This approach is a significant advancement for improved performance in NLP applications.

2.1.2.1 Word2Vec

Word2Vec was proposed and developed by Google in 2013 [7]. This model is built using two hidden layers in a shallow neural network. The neural network's output is a vector for each word. Syntactic and Semantic meaning of the word is supposed to be captured by the Continuous Bag of words (CBOW) and Skip-gram models.

Continuous Bag of Words (CBOW): CBOW predicts the word based on its context. Context is captured by communicating with neighbouring words.
Skip-Gram: This approach is the reverse of CBOW. The word is predicted using the context of the central word in skip-gram.

2.1.2.2 Global Vectors (GloVe)

Word2Vec captures the semantics of the words. However, it primarily focuses on the local context since it communicates only with the neighbouring words. GloVe is proposed to utilise the global statistic [8].

GloVe is dependent on a token in a corpus. This has mainly two steps.

1. Formation of the co-occurrence matrix for the corpus
2. Factorisation to get vectors

GloVe and Word2Vec are excellent in performance, accuracy and semantic understanding of the text in the large corpus. However, these models cannot work with out-of-vocabulary words.

2.1.2.3 FastText

Previously mentioned word embedding techniques had several issues such as longer training time and cannot perform in out-of-the-vocabulary words. FastText is proposed to address these limitations [9]. Instead of feeding a single word at a time, FastText feeds n-grams. N-grams allow the neural network to learn the relationship between words and semantics. Thus, the FastText provides better results, especially for the rare words. Facebook AI research released pre-trained word embedding for 157 languages [10], including Tamil and Sinhala. 157 FastText language models were trained using publically available Wikipedia articles. Fayaza and Ranathunga [3] have used FastText to embed Tamil news articles for the news clustering.

Although the models mentioned above perform well in understanding the syntax and semantics of a document, they do not capture the full context of the document required for many NLP tasks.

2.1.3 Contextual Embeddings

Contextual embedding was possible using Bi-Directional LSTM models. This section explores a few popular contextual embedding models.

Generic Context word representation (Context2Vec)

Melmbud et al. [11] proposed Context2Vec in 2016 to generate context-dependent representation. Context2Vec is again rooted on Word2Vec and CBOW, but instead of average word representation, Bi-directional LSTM was used.

Contextualised word representations Vectors (CoVe)

McCan et al. [12] proposed CoVe in 2017. The CoVe is based on Context2Vec, but instead of Word2Vec inside, they have used two layers of pre-trained BiLSTM for attention to seq2seq translation.

Embedding from Language Models (ELMo)

Petars et al. [13] introduced ELMo. With deep contextualised word representations, researchers hope to address the fluid behaviour of word use in grammar and semantics. The word representation is based on the bi-directional language model where ELMo gets the linear concatenation of both outputs. Linear concatenation make that same word can have two vector representations [19].

2.1.4 Transformer-based Pre-trained Language Models

Transformers have proven that they perform more efficiently and faster than LSTM or CNN models in language modelling. This subsection explores Transformer-based language models.

2.1.4.1 GPT (OpenAI Transformer)

Generative Pre-Training is the first Transformer-based language model. Radford et al. [15] proposed GPT in 2018. Unlike ELMo, GPT models the language using the Transformer's decoder. The decoder is an auto-regressive model that predicts the next word and outperforms GPT in left-to-right word prediction. GPT has the limitation of being a unidirectional model.

2.1.4.2 Bidirectional Encoder Representation from Transformers (BERT)

Google Language AI proposed BERT in 2018 [14, 16], and it has revolutionised the field of NLP ever since. BERT is the next generation of GPT, where language modelling is done on the large free text, and then the model is fine-tuned for specific tasks. Instead of sequential recurrence, BERT employs Transformer Neural network architecture with parallel attention layers. BERT is trained for the following two tasks to promote bidirectional text prediction and sentence-level understanding:

1. Masked Language Model (MLM) - 15% of the tokens are randomly masked, and the model predicts the masked token.
2. Next Sentence Prediction (NSP) - Pair of sentences are provided to the model, and BERT predicts the order of the sentence.

BERT is trained using the English book corpus and English Wikipedia text passages, and it is available as BERT-Base and BERT-Large. These models can be trained further and fine-tuned for specific tasks.

Subsequently, many researchers have proposed various variants of BERT for specific tasks. Some of the noteworthy variants are GPT2, XLNet, RoBERTa, ALBERT, XLM and DistilBERT.

2.1.4.3 BART

Lewis et al. [57] introduced Bidirectional and Auto-Regressive Transformers (BART). Part of the training text is corrupted and the objective of the model is to learn and reconstruct the text. Therefore, it is naturally effective in text generation and comprehension tasks. BART uses the seq-2-seq Transformer architecture. Text transformation for the model training was done using multiple techniques such as token masking, token deletion, token infilling, sentence permutation and document rotation.

Key differences between BERT and BART are that, BART's each layer of decoder additionally performs cross-attention over the final hidden layer and BART does not use additional feed-forward network before the word prediction. It achieves similar results as RoBERTa on discriminative tasks and better performance in text generation tasks [57].

2.1.4.4 Text-to Text Transfer Transformer (T5)

Google Research proposed T5 in 2020 [17]. T5 aims to understand and generate the language by transforming the data into the text-to-text format. Pre-training, corpus and model architectures are systematically contrasting from the previous models.

Although these models were able to address most limitations such as syntactic, semantic and contextual understanding of texts, they perform exceptionally well in specific domains or tasks. After training with targeted data and fine-tuning, a new variant of existing models should be proposed for each domain [19]. COVID Twitter BERT (CT-BERT) is one such example [18]. Table 1 shows the comparison of popular Language Models.

Table 1.
A comparison of popular Language Models
(Taken from [19])

LMs	Release Date	Architecture	Pre-Training Task	Corpus Used
Word2Vec	Jan-13	FCNN	-	Google News
GloVe	Oct-14	FCNN	-	Common Crawl corpus
FastText	Jul-16	FCNN	-	Wikipedia
ELMo	Feb-18	BiLSTM	BiLM	Wiki-Text-103
GPT	Jun-18	Transformer Decoder	LM	Book-Corpus
GPT-2	Jun-18	Transformer Decoder	LM	Web-Text
BERT	Oct-18	Transformer Encoder	MLM & NSP	WikiEn+Book-Corpus
RoBERTa	Jul-19	Transformer Encoder	MLM & NSP	Book-Corpus + CC-News +Open-Web-Text+ STORIES
ALBERT	Sep-19	Transformer Encoder	MLM+SOP	same as BERT
XLNet	Jun-19	Auto-regressive Transformer Encoder	PLM	WikiEn+ Book-Corpus+Giga5 + Clue-Web + Common Crawl
ELECTRA	2020	Transformer Encoder	RTD+MLM	same as XLNet
UniLM	2020	Transformer Encoder	MLM+NSP	WikiEn + Book-Corpus
MASS	2020	Transformer	Seq2Seq MLM	*Task-dependent
BART	2020	Transformer	DAE	same as RoBERTa
T5	2020	Transformer	Seq2Seq MLM	Colossal Clean Crawled Corpus (C4)

2.2 Multilingual Language Modelling (MLLM)

BERT has proven to be a success in the English language. Hence many language-specific BERT models were created using the same recipe, such as FlauBERT (French) [20]. However, these kinds of language-specific models are possible only for languages with adequate data and computational resources. The above limitation led research on "*How do we bring the benefit of such pre-trained BERT based models to a very long list of languages of interest?*" [20].

The goal of an MLLM is to learn a model that can support multilingual text embedding in the common vector space. Similar sentences with similar contexts from different languages should have a similar vector. mBERT and XLM-R are such MLLMs [20]. This section discusses the architecture of MLLM in detail.

2.2.1 Architecture

MLLMs are primarily based on the Transformer architecture discussed in section 2.1.4. MLLM includes three layers as follows: Input layer, Transformer layers and Output layer.

Input layer - Input is a sequence of tokens, and tokens are generated through a one-hot representation approach for finite vocabulary. Vocabulary from different languages is joined using algorithms such as Byte-Pair Encoding (BPE), wordpiece

and sentencePiece. Exponential weights are given to the vocabulary to ensure a reasonable representation [20].

Transformer layers - A typical MLLM contains the encoder of the Transformer network. It has *N* layers, with each containing *k* attention heads. Each attention head calculates the weighted embedding for each input token. This calculation includes the weight of other input tokens from the given sentence as well. The concatenated embedding of each token from each layer will be sent through a feed-forward neural network to produce a *d* dimensional embedding for each input token [20]. Table 2 shows the *N*, *k* and *d* parameters for a few MLLM models.

Table 2.
A comparison of existing Multilingual Language Models
(Taken from [20])

Model	Architecture				pretraining				Languages	
	N	*k*	*d*	*#Params.*	*Objective Function*	*Mono.*	*Parallel*	*Task specific data*	*#langs.*	*vocab.*
IndicBERT (Kakwani et al., 2020)	12	12	768	33M	MLM	IndicCorp	✗	✗	12	200K
Unicoder (Huang et al., 2019)	12	16	1024	250M	MLM, TLM, CLWR, CLPC, CLMLM	Wikipedia	✓	✗	15	95K
XLM-15 (Conneau and Lample, 2019)	12	8	1024	250M	MLM, TLM	Wikipedia	✓	✗	15	95K
XLM-17 (Conneau and Lample, 2019)	16	16	1280	570M	MLM, TLM	Wikipedia	✓	✗	17	200K
MuRIL (Khanuja et al., 2021)	12	12	768	236M	MLM, TLM	CommonCrawl + Wikipedia	✓	✗	17	197K
VECO-small (Luo et al., 2021)	6	12	768	247M	MLM, CS-MLM[†]	CommonCrawl	✓	✗	50	250K
VECO-Large (Luo et al., 2021)	24	16	1024	662M	MLM, CS-MLM	CommonCrawl	✓	✗	50	250K
InfoXLM-base (Chi et al., 2021a)	12	12	768	270M	MLM, TLM, XLCO	CommonCrawl	✓	✗	94	250K
InfoXLM-Large (Chi et al., 2021a)	24	16	1024	559M	MLM, TLM, XLCO	CommonCrawl	✓	✗	94	250K
XLM-100 (Conneau and Lample, 2019)	16	16	1280	570M	MLM, TLM	Wikipedia	✗	✗	100	200K
XLM-R-base (Conneau et al., 2020a)	12	12	768	270M	MLM	CommonCrawl	✗	✗	100	250K
XLM-R-Large (Conneau et al., 2020a)	24	16	1024	559M	MLM	CommonCrawl	✗	✗	100	250K
X-STILTS (Phang et al., 2020)	24	16	1024	559M	MLM	CommonCrawl	✗	✓	100	250K
HiCTL-base (Wei et al., 2021)	12	12	768	270M	MLM, TLM, HICTL	CommonCrawl	✓	✗	100	250K
HiCTL-Large (Wei et al., 2021)	24	16	1024	559M	MLM, TLM, HICTL	CommonCrawl	✓	✗	100	250K
Ernie-M-base (Ouyang et al., 2021)	12	12	768	270M	MLM, TLM, CAMLM, BTMLM	CommonCrawl	✓	✗	100	250K
Ernie-M-Large (Ouyang et al., 2021)	24	16	1024	559M	MLM, TLM, CAMLM, BTMLM	CommonCrawl	✓	✗	100	250K
mBERT (Devlin et al., 2019)	12	12	768	172M	MLM	Wikipedia	✗	✗	104	110K
Amber (Hu et al., 2021)	12	12	768	172M	MLM, TLM, CLWA, CLSA	Wikipedia	✓	✗	104	120K
RemBERT (Chung et al., 2021a)	32	18	1152	.559M[‡]	MLM	CommonCrawl + Wikipedia	✗	✗	110	250K

Output layer - It contains a superficial linear transformation layer followed by a softmax layer. The output layer produces a probability distribution for the input token over the tokens in the vocabulary. The output layer is only required during the pre-training phase and can be removed during the fine-tuning and task-inference phases. RemBERT has utilised this approach to reduce the model size [20, 21].

2.2.2 Training Objective Functions

Masked Language Model (MLM) and **Casual Language Model (CLM)** are the mostly used in monolingual models, as discussed in section 2.1.4.2. MLM and CLM both are unsupervised training functions. This section covers the Training Objective functions specialised for the MLLMs.

2.2.2.1 Parallel-Corpora based objective functions

Parallel-corpora objectives are designed and used in MLLM to have a similar vector for similar text across languages in the multilingual encoder space.

Translation Language Model (TLM) - Lample et al. [26], have introduced *supervised* TLM along with the cross-lingual model XLM. Two similar sentences (sequences) in different languages are fed into the model separated by the *[sep]* token. Random words in both sequences are masked with the *[mask]* token. The model needs to predict the masked words either using the first sequence context or otherwise from the context of the second sequence translation. The objective of TLM is similar to MLM; to minimise the cross-entropy loss of the masked tokens, but here the masked tokens can be in two different languages. TLM enforces the model to learn word alignments in both languages. [20]. Fig.1 depicts how TLM and MLM works.

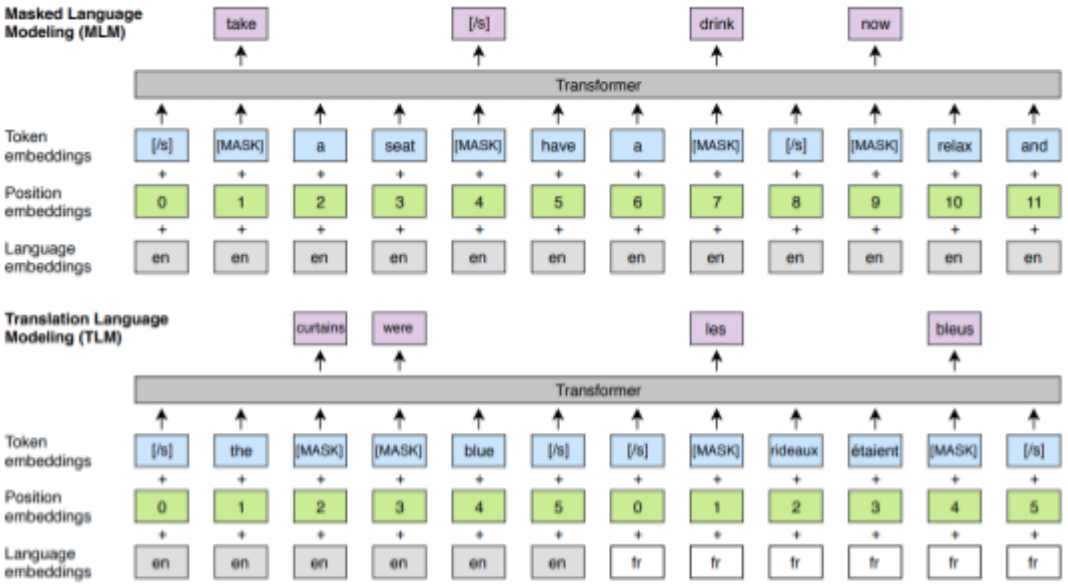


Fig.1. Cross-lingual language model pre-training. This compares the difference between MLM and TLM (Taken from [26])

Cross-attention Masked Language Modeling (CAMLM) - Ouyang et al. [20] proposed CAMLM. This approach is similar to TLM, but in contrast, CAMLM is restricted to use only the parallel sequence to predict the masked terms in the source sequence.

Cross-lingual Masked Language Modeling (CLMLM) - CLMLM is also similar to TLM. However, instead of using parallel sentences, CLMLM uses the input at the document level.

Following are few other examples of parallel-corpora based objective functions: Cross-lingual contrastive learning (XLCO), Hierarchical Contrastive Learning (HICTL) and Cross-lingual sentence alignment (CLSA).

2.2.2.2 Objective functions based on other parallel resources

In addition to parallel corpora, these objective functions use additional parallel resources like word alignments.

Cross-lingual word recovery (CLWR) - A matrix represents source language word embedding by the target language embeddings. The goal is to reconstruct the source language embeddings from the matrix.

Few other examples are Cross-Lingual paraphrase classification (CLPC), Alternating Language Model (ALM), Cross-lingual word alignment (CLWA) and Back translation masked language modelling (BTMLM). [20]

2.2.3 Pre-training data and Languages

MLLMs use two different sources of data.

1. Large monolingual corpora for each language
2. Parallel corpora between some languages

The source of data may differ for each model. For example, mBERT uses Wikipedia data, whereas XLM-R uses a common-crawl corpus. Training data for each model is listed in Table 2. Training with multilingual datasets will lead to data imbalance due to high resource languages like English. In order to avoid under-representation of low resource languages, MLLMs use an exponentially smoothed weighting of the data while creating pre-training data. (i.e. High resource languages will be undersampled, and low resource languages will be oversampled based) [20].

2.2.3.1 Curse of Multilinguality

Model capacity (number of parameters) is limited due to practical reasons. Therefore, the number of parameters allocated per language also will decrease with the introduction of new languages. Low-resource language performance can be increased when introducing similar high-resource languages until the capacity dilution kicks in. Therefore, as a summary, The curse of multilinguality degrades performance in all languages [27].

2.2.4 MLLM Models

This section covers a few renowned Multilingual and Cross-lingual language models.

2.2.4.1 LASER

Artetxe et al. [38] propose a single BiLSTM encoder-based model that has been pre-trained for 93 languages. According to the authors, LASER accomplishes impressive results without task-specific fine-tuning in contrast to mBERT. Model is trained using the publicly available parallel corpus OPUS [36]. The experiments show that LASER is effective in bitext mining, Cross-Lingual document classification and XNLI. Tamil and Sinhala are included in the pre-trained languages.

2.2.4.2 Multilingual BERT (mBERT)

mBERT is a 12 layer Transformer (Table 2) trained on the Wikipedia pages of 104 languages, including English, Tamil but not Sinhala. mBERT is available open-source for research [29].

Although mBERT is not explicitly trained for multilingual representation (mBERT uses monolingual training objective function MLM - Table 2 [20]), generalising cross-linguality has surprised the researchers. Piress et al. [25] hypothesise that having word pieces that must be mapped to communal space in all language families (numbers, URLs) forces the co-occurring pieces to be mapped to shared space. This extends the effect to all other word fragments. Fig.2. depicts the similarity of representation in two languages (i.e. 100 percent on EN-RU would mean that English and Russian are aligned to the exact representation). Authors say that the accuracy decreases over the last few layers, most likely because the model requires further language-specific data to correctly fill in the missing word [25].

Wu et al. [44] compare developing cross-lingual structures in pre-trained MLLMs. They claim that MLLMs such as mBERT and XLM enable effective cross-lingual transfer. This means the mentioned language models are able to learn from supervised data for one language and implement it to the next without extensive training. Following are few hypothesised factors for mBERT to be multilingual

- Domain similarity
- Anchor points (Shared vocabularies across languages)
- Parameter sharing
- Language similarity

However, they found that the most important factor is parameter sharing. Joint BPE and domain similarity contribute a little in learning cross-lingual representation and anchor points are not necessarily effective in cross-lingual transfer.

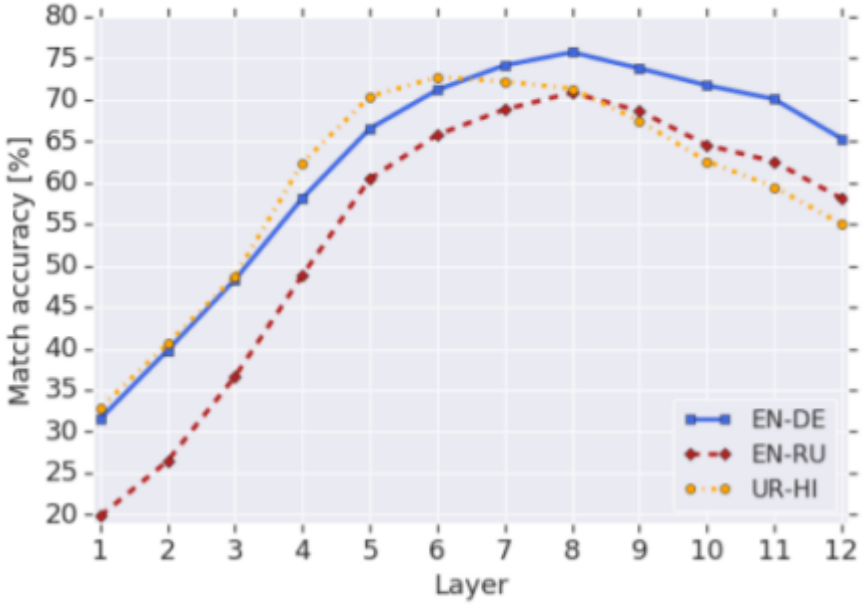


Fig.2 M-BERT Accuracy of nearest neighbour translation for EN-DE, EN-RU and HI-UR (Taken from [25])

2.2.4.3 XLM (Cross-lingual Language Model)

Lample et al. [26] proposed XLM with a new objective function, Translation Language Model (TLM) to provide the cross-lingual capability. Section 2.2.2.1 and Fig.7 explain the TLM and the difference between traditional MLM and TLM. XLM beats mBERT performance in the XNLI dataset by 4.9% after including both the MLM and TLM in the training objective function. Table 3 shows that this model achieves on average 75.1% for the cross-lingual sentence encoder. This handles only 15 languages [27].

Table 3
XLM results on cross-lingual classification accuracy
(Taken from [26])

	en	fr	es	de	el	bg	ru	tr	ar	vi	th	zh	hi	sw	ur	Δ
Machine translation baselines (TRANSLATE-TRAIN)																
Devlin et al. [14]	81.9	-	77.8	75.9	-	-	-	-	70.7	-	-	76.6	-	-	61.6	-
XLM (MLM+TLM)	85.0	80.2	80.8	80.3	78.1	79.3	78.1	74.7	76.5	76.6	75.5	78.6	72.3	70.9	63.2	76.7
Machine translation baselines (TRANSLATE-TEST)																
Devlin et al. [14]	81.4	-	74.9	74.4	-	-	-	-	70.4	-	-	70.1	-	-	62.1	-
XLM (MLM+TLM)	85.0	79.0	79.5	78.1	77.8	77.6	75.5	73.7	73.7	70.8	70.4	73.6	69.0	64.7	65.1	74.2
Evaluation of cross-lingual sentence encoders																
Conneau et al. [12]	73.7	67.7	68.7	67.7	68.9	67.9	65.4	64.2	64.8	66.4	64.1	65.8	64.1	55.7	58.4	65.6
Devlin et al. [14]	81.4	-	74.3	70.5	-	-	-	-	62.1	-	-	63.8	-	-	58.3	-
Artetxe and Schwenk [4]	73.9	71.9	72.9	72.6	73.1	74.2	71.5	69.7	71.4	72.0	69.2	71.4	65.5	62.2	61.0	70.2
XLM (MLM)	83.2	76.5	76.3	74.2	73.1	74.0	73.1	67.8	68.5	71.2	69.2	71.9	65.7	64.6	63.4	71.5
XLM (MLM+TLM)	**85.0**	**78.7**	**78.9**	**77.8**	**76.6**	**77.4**	**75.3**	**72.5**	**73.1**	**76.1**	**73.2**	**76.5**	**69.6**	**68.4**	**67.3**	**75.1**

2.2.4.4 XLM-R (XLM-RoBERTa)

Conneau et al. [27] propose XLM-R, a transformer-based multilingual masked language model. Authors have observed that fine-tuned unsupervised masked language modelling (MLM) training objective has achieved more than 75% average accuracy, which was on par with XLM's supervised objective function - translation language modelling (TLM). Therefore, they have decided not to use the supervised TLM objective function.

They have used 2.5TB CommonCrawl corpus in 100 languages instead of Wikipedia corpora since Wikipedia is too small to enable unsupervised representation learning. XLM-R outperformed state-of-the-art models like XLM and Unicoder and obtained 80.9% accuracy (Table 4).

Table 4
XLM-R Results on cross-lingual classification in XNLI
(Taken from [27])

Model	D	#M	#g	en	fr	es	de	el	bg	ru	tr	ar	vi	th	zh	hi	sw	ur	Avg
Fine-tune multilingual model on English training set (Cross-lingual Transfer)																			
Lample and Conneau (2019)	Wiki+MT	N	15	85.0	78.7	78.9	77.8	76.6	77.4	75.3	72.5	73.1	76.1	73.2	76.5	69.6	68.4	67.3	75.1
Huang et al. (2019)	Wiki+MT	N	15	85.1	79.0	79.4	77.8	77.2	77.2	76.3	72.8	73.5	76.4	73.6	76.2	69.4	69.7	66.7	75.4
Devlin et al. (2018)	Wiki	N	102	82.1	73.8	74.3	71.1	66.4	68.9	69.0	61.6	64.9	69.5	55.8	69.3	60.0	50.4	58.0	66.3
Lample and Conneau (2019)	Wiki	N	100	83.7	76.2	76.6	73.7	72.4	73.0	72.1	68.1	68.4	72.0	68.2	71.5	64.5	58.0	62.4	71.3
Lample and Conneau (2019)	Wiki	1	100	83.2	76.7	77.7	74.0	72.7	74.1	72.7	68.7	68.6	72.9	68.9	72.5	65.6	58.2	62.4	70.7
XLM-R$_{Base}$	CC	1	100	85.8	79.7	80.7	78.7	77.5	79.6	78.1	74.2	73.8	76.5	74.6	76.7	72.4	66.5	68.3	76.2
XLM-R	CC	1	100	**89.1**	**84.1**	**85.1**	**83.9**	**82.9**	**84.0**	**81.2**	**79.6**	**79.8**	**80.8**	**78.1**	**80.2**	**76.9**	**73.9**	**73.8**	**80.9**
Translate everything to English and use English-only model (TRANSLATE-TEST)																			
BERT-en	Wiki	1	1	88.8	81.4	82.3	80.1	80.3	80.9	76.2	76.0	75.4	72.0	71.9	75.6	70.0	65.8	65.8	76.2
RoBERTa	Wiki+CC	1	1	**91.3**	82.9	84.3	81.2	81.7	83.1	78.3	76.8	76.6	74.2	74.1	77.5	70.9	66.7	66.8	77.8
Fine-tune multilingual model on each training set (TRANSLATE-TRAIN)																			
Lample and Conneau (2019)	Wiki	N	100	82.9	77.6	77.9	77.9	77.1	75.7	75.5	72.6	71.2	75.8	73.1	76.2	70.4	66.5	62.4	74.2
Fine-tune multilingual model on all training sets (TRANSLATE-TRAIN-ALL)																			
Lample and Conneau (2019)†	Wiki+MT	1	15	85.0	80.8	81.3	80.3	79.1	80.9	78.3	75.6	77.6	78.5	76.0	79.5	72.9	72.8	68.5	77.8
Huang et al. (2019)	Wiki+MT	1	15	85.6	81.1	82.3	80.9	79.5	81.4	79.7	76.8	78.2	77.9	77.1	80.5	73.4	73.8	69.6	78.5
Lample and Conneau (2019)	Wiki	1	100	84.5	80.1	81.3	79.3	78.6	79.4	77.5	75.2	75.6	78.3	75.7	78.3	72.1	69.2	67.7	76.9
XLM-R$_{Base}$	CC	1	100	85.4	81.4	82.2	80.3	80.4	81.3	79.7	78.6	77.3	79.7	77.9	80.2	76.1	73.1	73.0	79.1
XLM-R	CC	1	100	**89.1**	**85.1**	**86.6**	**85.7**	**85.3**	**85.9**	**83.5**	**83.2**	**83.1**	**83.7**	**81.5**	**83.7**	**81.6**	**78.0**	**78.1**	**83.6**

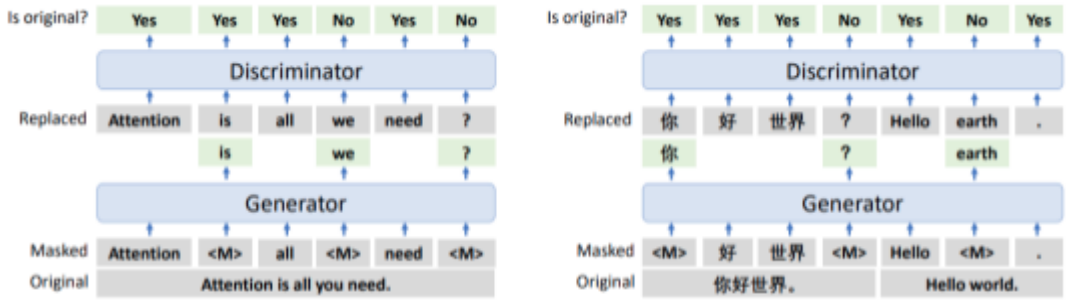
Authors have found that their MLLMs can outperform their monolingual BERT counterparts. XLM-R is pre-trained for both Tamil and Sinhala, and the model is available for public use.

2.2.4.5 XLM-E (XLM-ELECTRA)

Chi et al. [28] propose XLM-E in 2021, and it is at the top of the XTREME leaderboard (Fig.5) as of September 28th of 2021. ELECTRA contains two Transformer encoders that serve as Generator and Discriminator.

1. Generator - Takes masked sentences as input and predict the masked tokens. The output is fed into the discriminator.
2. Discriminator - This takes corrupted sentences from the generator as input and predicts if the generator replaces the token.

Altogether XLM-E pre-training is focused on reducing the loss in MLM, TLM, MRTD and TRTD (Fig.3). Fig.3 shows pre-training tasks of XLM-E. Authors claim that XLM-E uses only 1% of the computation power required by XLM-R.



(a) Multilingual replaced token detection (MRTD) (b) Translation replaced token detection (TRTD)

Fig.3 Overview of two pre-training tasks of XLM-E
(Taken from [28])

Although XLM-E outperforms XLM-R_base by 0.3%, it does not match XLM-R's performance (80.9%) in the XNLI classification. XLM-E uses the same 100 languages CommonCrawled data by Conneau et al. [27]. However, the model and the code is not open-sourced yet.

2.2.5 MLLM Benchmarks

The most common approach to evaluate the MLLM is to find the cross-lingual performance on downstream tasks. MLLM is fine-tuned for the specific task in a high resource language such as English, and then it is evaluated against other languages. XGLUE, XTREME and XTREME-R are few standard benchmarks [20]. Following are a few downstream tasks that are evaluated in XTREME and XGLUE.

1. **Classification** - Here, the task is to classify the input into k classes. XNLI is one popular dataset used for classification [20].
2. **Structure prediction** - Here, the task is to predict the label for every word in the input sequence.
3. **Question answering** - Here, the task is to find the answer from the given sentence context. XQuAD is the most commonly used dataset.
4. **Retrieval** - Here, the task is to retrieve matching sentences from the collection of target language given the sentence in a source language. BUCC and Tateoba are the most commonly used datasets.

Cross-lingual TRansfer Evaluation of Multilingual Encoders (XTREME) was created to promote more study on multi-lingual transfer learning. XTREME comprises eight tasks and includes 40 typologically different languages from 12 language groups.

Languages for the evaluation are carefully selected to maximise the language diversity, coverage and availability of the training data. Tamil, Telugu and Malayalam are some of the languages in XTREME evaluation [23, 24]. Table 27 Shows the current XTREME benchmark leaderboard standings. Microsoft's Turing ULR V5 tops the ranking followed by XLM-R is in 12th, mBERT is in 14th and RemBERT in 16th.

Table 27. XTREME Leaderboard summary as of 2021 September 25th
(Taken from [23])

Rank	Model	Participant	Affiliation	Attempt Date	Avg	Sentence-pair Classification	Structured Prediction	Question Answering	Sentence Retrieval
0		Human	-	-	93.3	95.1	97.0	87.8	-
1	Turing ULR v5	Alexander v-team	Microsoft	Sep 17, 2021	83.7	90.0	81.4	74.3	93.7
2	VECO	DAMO AliceMind Team	Alibaba	Sep 21, 2021	82.0	89.0	76.7	73.4	93.3
3	CoFe	HFL	iFLYTEK	Sep 21, 2021	81.9	88.4	75.7	73.9	93.5
4	Polyglot	MLNLC	ByteDance	Apr 29, 2021	81.7	88.3	80.6	71.9	90.8
5	Unicoder + ZCode	MSRA + Cognition	Microsoft	Apr 26, 2021	81.6	88.4	76.2	72.5	93.7
6	ERNIE-M	ERNIE Team	Baidu	Jan 1, 2021	80.9	87.9	75.6	72.3	91.9
7	HICTL	DAMO MT Team	Alibaba	Mar 21, 2021	80.8	89.0	74.4	71.9	92.6
8	T-ULRv2 + StableTune	Turing	Microsoft	Oct 7, 2020	80.7	88.8	75.4	72.9	89.3
9	Anonymous3	Anonymous3	Anonymous3	Jan 3, 2021	79.9	88.2	74.6	71.7	89.0
10	FILTER	Dynamics 365 AI	Microsoft	Sep 8, 2020	77.0	87.5	71.9	68.5	84.4

2.2.6 Multilingual Knowledge Distillation

Reimers et al. [33] propose a new approach to efficiently map languages in the common vector space using the teacher-student model. The authors discovered that this approach outperforms LASER by up to 40% in accuracy for low-resource languages. They claim that the problem with XLM-R and mBERT is that they produce poor text representation out of the box. Furthermore, the vector spaces of the different languages are not matched [34, 33]. (In other words, phrases with the same content in various languages are plotted to various spots in the vector space.). Authors have used English SBERT (Sentence-BERT) as the teacher model and XLM-R as the student model. Fig.5 shows how the teacher-student model works. The student model learns multilingual sentence vector space with 2 critical properties

1. Across languages, vector space is aligned (i.e., sentences with similar content should be in different languages should be plotted closely)

2. Vector space characteristics from the teacher model are embraced and transmitted to other languages using the actual source language.

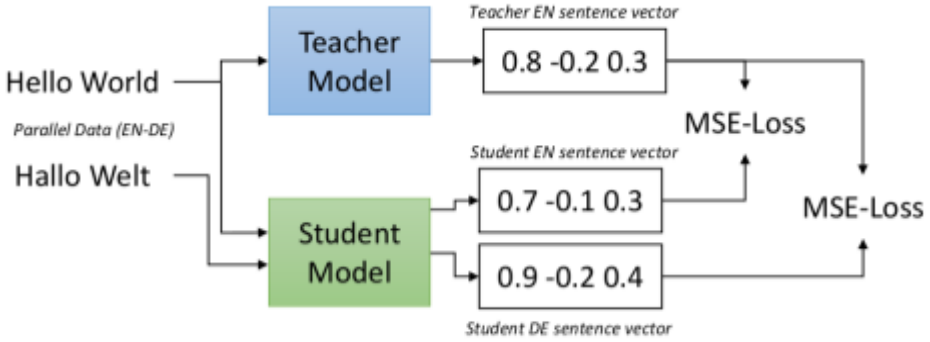


Fig.5. Given parallel data (e.g. English and German), train the student model such that the produced vectors for the English and German sentences are close to the teacher English sentence vector.
(Taken from [33])

According to Reimers et al. [33] MLLMs perform competitively in monolingual tasks, However, in multilingual experiments, the results are significantly lower. Authors say that it can be because the vector space is not aligned in mBERT and XLM-R across languages. Note that mBERT and XLM-R are not fine-tuned for the tasks here.

2.3 Document Clustering

The process of organising text articles into groups or clusters is known as document clustering. Documents that belong to a cluster are about the same subject, whereas documents belong to another cluster are about different subjects. This has to be done automatically without manual intervention.

2.3.1 Clustering approaches

Saiyad et al. [1] summarise the document clustering approaches and problems before the BERT era. The traditional approach is to use a bag of words with co-occurrence frequency. The drawback is that the traditional approach does not take the semantic representation of the words. Following are the challenges and problems in implementing Document Clustering.

- Selection of appropriate similarity measure
- Selection of appropriate clustering technique
- Finding appropriate evaluation measures
- Synonymy and Polysemy - Since this research will use pre-trained models, it will not be a problem.
- High dimensionality
- Cluster labelling

- Selection of appropriate features for clustering - Since this research will be focused on word embedding, this will not be a problem.

To evaluate results, a few metrics such as F-measure, silhouette coefficient, precision, recall, and entropy are widely utilised. Cosine, inter-similarity and Euclidean distance are the commonly used similarity measures. Table 5 shows the common clustering algorithms used in document clustering in the literature.

Table 5
Comparison of clustering algorithms used in document clustering
(source: [1])

Clustering Algorithm	**Advantages**	**Disadvantages**
K-Means	- Produces hard clusters, particularly if the clusters are round.	- The value of k is difficult to predict. - Do not overlap
Bisecting k-means	- The cost of clustering is low, and the quality of clustering is very good. - No need to mention K	- NIL
Self-organising map	- Data mapping is simple to depict. - Has the ability to organise large amounts of data.	- Difficulty deciding on input weights - Mapping results into divided clusters
Particle swarm optimisation	- More stable - Optimization is possible	- Too slow - Non-repeatable due to computation costs
Hierarchical agglomerative clustering	- Easy to implement - No of cluster info is not needed - The cluster quality is adequate.	- Computationally cost - The denogram makes it difficult to determine the exact number of clusters.
CLARANS		- Useful for databases
DBSCAN	- Works faster and end up with better final results	- Has difficulty in distinguishing separated clusters if they are located too

		close
Fuzzy c-means algorithm	- Greater outcomes for overlapping data and comparable to k-means	- Number of clusters needs to be provided

One pass algorithm - Leban et al. [43] proposed (Fig.6) to cluster documents based on their similarity, and indeed the number of clusters would be dynamically determined based on the semantic information.

a) Assign the first article to the first cluster and make its centroid equal to article's values.

b) For each remaining article of the article list,

Calculate the similarity with each of the current clusters as: *Similarity (article, cluster centroid)*

- Determine the maximum similarity(Similarity Max) value of the above calculated similarity values for a particular article
- If Similarity Max >= THRESHOLD, add the current article to the cluster with maximum similarity value.
  - Update the cluster centroid (centroid value was calculated by taking mean value of all articles)
  - Determine the closest centroid to the updated centroid and merge them if their similarity is above a predefined threshold.
- If Similarity Max < THRESHOLD, make the current article a new cluster and update the clusters list.

Algorithm 1. One pass algorithm for article grouping

Fig.6 One pass Clustering Algorithm
(Taken from [4])

3.0 RELATED WORKS

This section covers the literature in document clustering where features are extracted from language models and word embedding models (Eg: BERT, FastText)

TF-IDF and monolingual BERT

Marcińczuk et al [30], Fayaza and Ranathunga [3] compared the TF-IDF approach against the modern approaches. Marcińczuk et al. experimented doc2vec and BERT in the Polish language documents and they observed that BERT obtained significantly lower results than other methods. In contrast, Fayaza and Ranathunga experimented with FastText and they found that FastText obtained better results than the TF-IDF approach. Marcińczuk believes that the reason for underperformance of the BERT is that its model takes only the first 512 tokens as input, and they suspected that might not be enough to depict the topic of the document. Also, they said that their BERT model is not fine-tuned for document clustering, but they have seen other researchers have obtained better results using fine-tuned BERT. M. F. Fayaza and S. Ranathunga also found that articles with titles and bodies gave better performance compared to just the title.

Gusev et al. [39] discuss language specific BERT model usage for Russian news clustering. They have hosted a Russian news clustering competition with crawled data from different sources. Authors say that most participants preferred to handle this problem as classification rather than as clustering. An ensemble of four BERT MLLMs trained with stochastic weight averaging produced the best results. A single RuBERT model finished second. The best clustering-based model finished fourth overall, used Global Multihead Pooling and contrastive loss and was based on S-BERT over RuBERT. Fourth place participants have published a paper about their methodology [40].

Multilingual language models for monolingual document clustering

Stankevičius et al. [32] compared MLLMs, mBERT and XLM-R against the doc2vec models for Lithuanian document clustering. This research did not include cross-lingual documents. They found that the Doc2Vec version easily outperformed MLLMs. mBERT and other pre-trained models are meant to be fine-tuned for such a task at hand. MLM fine-tuning on mBERT has been performed by authors with approximately 28000 articles. Although BERT can take upto 512 tokens as the upper limit, they found that 144 tokens yielded the best results. They also have considered experimenting with XLM-R. However, the massive size of the XLM-R model has limited the experiments, and authors were not able to further optimise the model due to the high computational cost. The Doc2Vec version easily outperformed MLLMs.

Authors were able to see the improvement in mBERT models after fine-tuning, but they were not sure how long further it needed to be fine-tuned.

Cross Lingual document clustering with MLLM and TF-IDF

Miranda et al. [42], Linger et al. [41], Wu et al. [44] and S.Dutta [37] worked on cross-lingual document clustering. Miranda et al. [42] used the TF-IDF approach whereas other authors used MLLMs. All of them used cosine value to measure the similarity between the documents, however, Miranda used English as the pivot language where distance for every other language is compared to English. This approach provided a competitive F1 score of 84.0. Linger used multilingual DistilBERT made with SBERT for batch clustering of multilingual news streaming. Wu et al. [44] trained monolingual BERT with 10k parallel sentences to do the cross-lingual sentence embedding. This gave good results, however, state of the art performance is above 95% and achieved by LASER (Fig.7). They have aligned sentences in monolingual BERT using the Procrustes approach with fastAlign. Dutta [37] compared mBERT, DistilBERT (with S-BERT) and a newly proposed Aligned Word Mover's Distance (WMD-A) word alignment algorithm. He found that DistilBERT and the WMD-A performed similarly and the mBERT performed poorly. Therefore, it can be concluded from the above-mentioned research that DistilBERT (with S-BERT) and LASER performs better than the mBERT. However, DistilBERT does not support Sinhala and Tamil languages out of the box; it needs to be trained with bi-texts. Also, XLM-R was not experimented in the cross-lingual document clustering. Few researchers have used word alignment techniques to improve the performance.

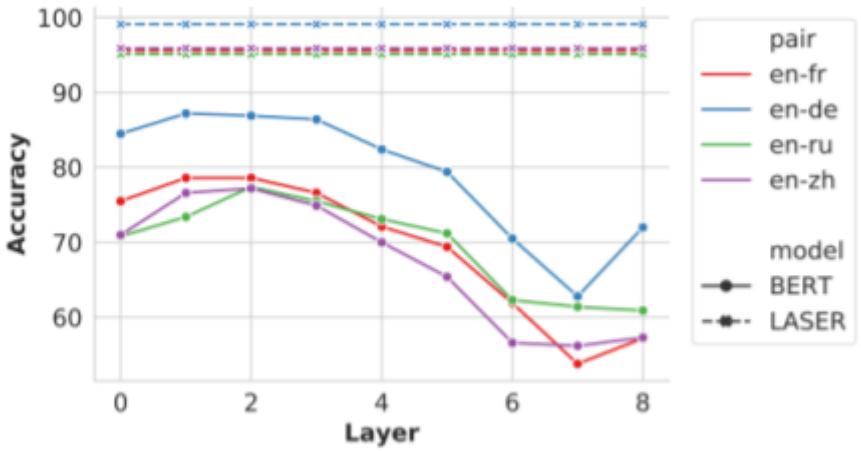


Fig.7. Parallel sentence retrieval accuracy after procrustes alignment of monolingual BERT
(Taken from [44])

Clustering

Fayaza and Ranathunga [3] have experimented with both affinity propagation algorithm and one-pass algorithm. They found that one pass algorithm outperformed the affinity propagation algorithm and the affinity propagation algorithm failed to find single document clusters. Stankevičius et al. [32] sampled 1500 equally distributed articles in 50 clusters. Since the cluster number is known, they used the k-means clustering algorithm. Miranda et al. [42] and Park et al. [31] developed a novel clustering algorithm. Miranda et al.'s algorithm clusters documents in monolingual space into cross-lingual space. Park [31] says that the downside of the BERT is the dimensionality of the contextualised embeddings. It generates 768 dimensions for a single word token. This can cause notorious issues in the computation of the similarity measure. Both the proposed algorithms use cosine similarity as a similarity measure. Table 6 shows the summary of clustering approaches conducted in different researches.

Table 6.
Summary of approaches conducted in the researches

Research	**Clustering method**	**Similarity measure**	**Evaluation**	**Dataset**
[30]	Agglomerative clustering algorithm	Cosine	Adjusted Mutual Information	Manually labelled three datasets (633, 362 and 2029 documents in each of 3 datasets)
[3]	Affinity propagation algorithm and One-Pass	Cosine	Pairwise F-Score against manual clustering and Kappa statistic	Crawled from 9 Tamil news sites.
[31]	Advanced document clustering (Proposed)	Cosine	Unsupervised clustering accuracy (ACC) and Normalised mutual information (NMI)	SQuAD 1.1 (10 largest among 536 articles), REUTERS (10000 news articles) and FakeNewsAMT (240 documents)

[32]	K-Means	Not mentioned	Matthews Correlation Coefficient (MCC)	1500 evenly distributed articles for 50 clusters

Vector alignment in multilingual word embedding

Wu et.al [45] experimented on whether explicit alignments improve multilingual encoders. According to the authors, mostly all encoders have been pre-trained with no explicit cross-lingual goals (i.e promoting words with similar context from different languages to have similar representations). Improvements can be made by using explicitly cross-lingually linked data during the pre pre-training phase (E.g bitext, dictionaries). Following alignment techniques have been experimented in XLM and mBERT.

- Linear Mapping
- L2 Align
- Contrastive alignment (Proposed)
    - Weak alignment
    - Strong alignment

They found that XLM-R Large performs much better than XLM-R Base with or without alignment. XLM-R Large has almost two fold input attributes as XLM-R Base but is trained on the same data. This implies that increasing model capacity makes it more possible to result in improved cross-lingual representation compared to employing explicit alignment objectives.

3.2 Summary

Early work on multilingual clustering used TF-IDF [42]. Then MLLMs were used in document clustering. However, their performance can be inferior without fine-tuning [32, 37]. MLLM must be used after the fine-tuning for downstream tasks [37, 33] (XLM-R fine-tuning requires a lot of computational resources [32]). Knowledge Distillation (Multilingual SBERT) with distillation provided better results compared to mBERT [33].

Researchers have used TF-IDF, doc2Vec as baseline but tend to perform better than MLLM in monolingual document clustering [30,32]. SBERT was found to be the

most preferred variant to embed documents and provides better and quicker results [37, 39, 41].

Different variations of input were used in the MLLM document clustering since the input token count is limited (512 for BERT). Reducing the input token count increased the performance [32, 40]. Title and body were concatenated [41] and experimented too.

It was found that vectors are not properly aligned in MLLMs across languages (Especially in low resource languages) [33]. Hence vector alignment is needed to boost the performance. Bi-text or dictionaries were also employed to boost the accuracy of cross-lingual transfer [45]. XLM-R Large performed better with or without explicit alignment [45]. Explicit vector alignment does not improve the performance in XLM-R Base [45]. Novel alignment approach also was discovered and named Word Mover's Distance [37].

Cosine has been used as a similarity measurement. One pass clustering algorithm tends to provide better results [3] (Traditional clustering algorithms may not work well due to the huge number of dimensions. BERT has 768 dimensions per token [31]. It should be PCA-ed before the clustering). Few researchers came up with different clustering approaches [42, 31] to tackle the cross vector space problem. F1, Matthews Correlation Coefficient, Kappa can be used for evaluating the clusters.

4.0. METHODOLOGY

This section covers the implementation details of the research.

4.1 System architecture

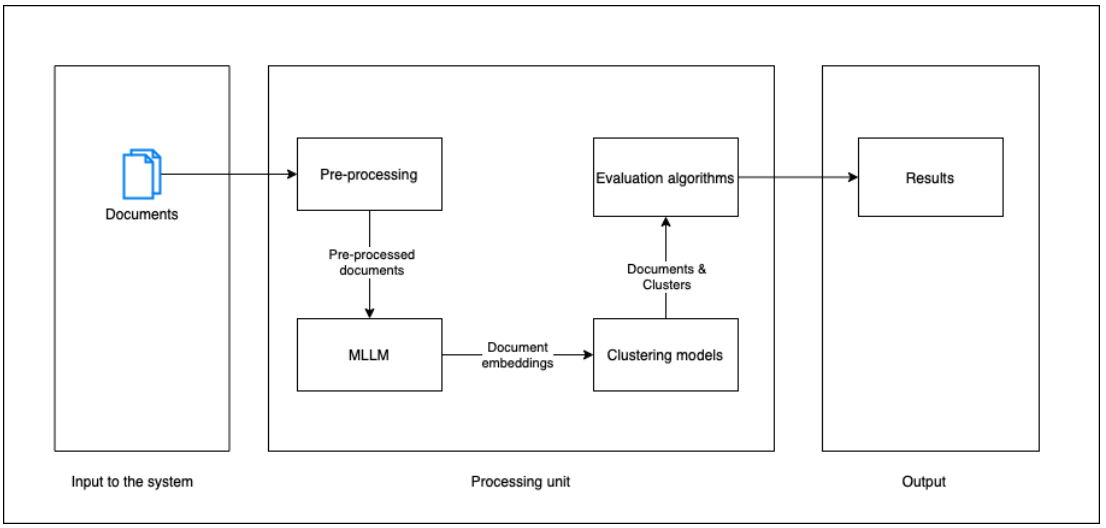


Fig.9 System architecture

Collected documents were pre-processed initially and then word embedding was obtained through the MLLMs. Then the documents were clustered using the K-Means and One-Pass algorithm. Fig.9 shows the system architecture.

4.2 Models

Previous researchers have experimented with XLM-R [32, 45], mBERT [32, 37, 45], LASER [44] and knowledge distillation [41] approaches. mBERT does not support Sinhala language and LASER is not a pre-trained Transformer-based MLLMs. Knowledge distillation is also not available out of the box for Tamil and Sinhala languages. It has to be trained using a huge parallel corpus. Therefore, this research was conducted using the XLM-R models. XLM-R Base and XLM-R Large were experimented in this research. Related packages were obtained from the HuggingFace transformer library.

4.2.1 Baseline

Researchers have used TF-IDF [3, 30], Doc2Vec [32] monolingual BERT [44] as the baseline. They have also found that non-contextual word embedding methods such as TF-IDF and Doc2Vec simply outperformed the MLLMs. This study experimented both TF-IDF and mBERT as the baseline with One-Pass clustering algorithm. Code

is available in the Github repository[1]. TF-IDF does not solve the research problem as discussed in the Evaluation section. Therefore mBERT is used as the baseline.

4.3 Clustering algorithms

Researchers have used Agglomerative clustering algorithm [30], Affinity propagation algorithm [3], One-Pass algorithm [3] and K-Means [32] in their studies of document clustering. However they all found that traditional clustering approaches do not well due to the high dimensional vectors that are produced by the word embedding. This study experiments both One-Pass and K-Means clustering algorithms.

Almost all the research [3, 30, 31] has used cosine to measure the similarity between the documents. This research also uses cosine similarity referred to in Algorithm 2.

One-Pass clustering Algorithm [43]

```
Assign the first document as the first cluster's centroid.
For all other documents:
        For all clusters:
                Find the cosine similarity between document and cluster centroid
        If the MAX cosine similarity > THRESHOLD:
                Assign the document to the respective cluster
                Update the cluster centroid as the mean
        Else:
                Add the document to new cluster
                Update the document vector as the centroid
```

Algorithm 1 One-Pass algorithm

K-Means

Sklearn implementation of K-Means was used in this work. Euclidean distance was used to identify the similarity among the documents.

Cosine similarity Algorithm

```
Get two vectors (a, b) that need similarity report

Find the similarity by getting the dot product as follow
        cos_sim = dot(a, b)/(norm(a)*norm(b))

Return the similarity cos_sim
```

Algorithm 2 Cosine similarity algorithm

[1] https://github.com/VIthulan/multilingual-doc-clustering/blob/main/src/Baseline/tf-idf-baseline.ipynb

4.4 Accuracy Algorithm

Past researchers have used Adjusted Mutual Information [30], Pairwise F-Score [3], Unsupervised clustering accuracy (ACC) and Normalised mutual information (NMI) [31], Matthews Correlation Coefficient (MCC) [32]. However, in this research the collected data can belong to multiple categories at the same time. Therefore the above evaluation metrics are not possible to use. Algorithm 3 shows the accuracy calculation algorithm [58].

```
For all clusters:
        Find the category by taking the majority
        Calculate the the purity of the cluster
Average the purity of all the clusters
```

Algorithm 3 Evaluation algorithm

4.5 Optimal length finding algorithm

Trimmed content was experimented as suggested in previous research [32, 40]. Algorithm 4 implemented by me shows the way to find the optimal length for better accuracy.

```
For content length in range (Min, Max):
        For each One-Pass threshold steps:
                Find accuracy
        Find max accuracy from all thresholds
Find max accuracy from all content length
Return max accuracy, content length and threshold
```

Algorithm 4 Optimal content length finding algorithm

4.6 Input variations

Following criteria were largely checked in different experiments

1. Document title only - Only document title was used in the word embedding.
2. Document content only - Only document body was used in the word embedding.
3. Document title and content - Both title and body were used when creating the word embeddings.
4. Trimmed content - Body was trimmed before it was used in the word embedding.
5. Title and trimmed content - Body was trimmed before it was used in the word embedding along with the title.

4.7 Implementation

Python was used as the programming language to implement the solution. mBERT was used as the baseline with a one-pass clustering algorithm. XLM-R supports English, Sinhala and Tamil and not many cross-lingual document clustering were not experimented with it. Therefore, we have decided to continue our research in XLM-R. Code is available in and Google Colab[2]. Accuracy algorithm was developed in Google Colab and available here[3].

[2] https://colab.research.google.com/drive/1VPdeNoTxQu9QZx9rBdvgdeSFKPWN2lLq?usp=sharing

[3] https://colab.research.google.com/drive/1qoGMjyiFmuC-ehWluOaxePKBOWV0FDex?usp=sharing

5.0 EVALUATION

5.1 Dataset

Past researchers have used manually collected and labelled data [30, 32], crawled data [9], SQuAD [31] in their studies. This study also collected 1604 Tamil, Sinhala and English documents manually and labelled it for the clustering and evaluations. Data quality also verified.

5.1.1 Data collection

Documents were collected manually and annotated from eighteen different sources. Table 7 below shows the data sources and the language they belong to.

Table 7.
Data sources where the data has been collected

Data source	URL	Si	En	Ta
News 1st Sinhala	https://www.newsfirst.lk/sinhala/	✓		
News 1st English	https://www.newsfirst.lk		✓	
News 1st Tamil	https://www.newsfirst.lk/tamil/			✓
Newswire	https://www.newswire.lk		✓	
Ada derana Sinhala	http://sinhala.adaderana.lk	✓		
Ada derana English	http://www.adaderana.lk		✓	
Ada derana Tamil	http://tamil.adaderana.lk			✓
BBC Sinhala	https://www.bbc.com/sinhala	✓		
BBC English	https://www.bbc.com/news/topics/cywd23g0gxgt/sri-lanka		✓	
BBC Tamil	https://www.bbc.com/tamil			✓
Sooriyan FM news	https://www.sooriyanfm.lk			✓
Veerakesari	https://www.virakesari.lk			✓
Dailymirror	https://www.dailymirror.lk		✓	
Island.lk	https://island.lk		✓	
Ceylon today	https://ceylontoday.lk		✓	

Hiru English	https://www.hirunews.lk/english/		✓	
Hiru news	https://www.hirunews.lk	✓		
Ada.lk	https://www.ada.lk	✓		

Documents were collected in a JSON format [46] so that further processing would be easier. Altogether 1604 documents were collected under this process. All the documents were committed to github and publicly available for future researchers [47]. The collected documents were created between January-March of 2022.

All the documents were categorised into at least one category out of ‘general', 'crime', 'political', 'business', 'economic', 'sports', 'arts', 'entertainment', 'education', 'tech', 'auto', 'legal', 'lifestyle', 'health', 'covid', 'weather'. These categories were identified as the major classes that a document can belong to from a survey done by Reuters Institute for the Study of Journalism [48] and a discussion from Quora [49].

Due to the nature of the documents, an article can belong to multiple categories as shown below.


```
{
  "title": "Power Cuts from today (18); PUSCL grants permission",
  "content": "The Public Utilities Commission of Sri Lanka has granted
permission to impose power cuts in Sri Lanka. Accordingly, a one-hour power cut
will be imposed between 2:30 PM to 6:30 PM under 4 separate groups. In addition,
a 45-minute power cut will be imposed between 6:30 PM to 10:30 PM as well.",
  "url":
"https://www.newsfirst.lk/2022/02/18/power-cuts-from-today-18-puscl-grants-permi
ssion/",
  "date": "2022-02-18",
  "category": "general, economy"
}
```


Categories were represented in a one-hot encoding method and all the collected documents were consolidated into one single CSV after multiple rounds of reviews by different people. Consolidated file has been committed to Github publicly for future research purposes[4].

[4] https://github.com/VIthulan/multilingual-doc-clustering/blob/main/src/processed_data/reviewed/documents_annotated_v2_1.csv

5.1.2 Data quality

Data quality has to be rigorously verified since these documents were manually collected. A script[5] was implemented and used to verify the following

1. All required fields should be available in the JSON document
2. All mandatory fields should not have empty values
3. Document content should be greater than the given value
4. Date should be in the given format
5. Category field should contain only allowed values
6. Should not contain duplicate documents

Invalid records that were identified by the script were excluded from the research implementation.

5.1.3 Inter annotator agreement (IAA)

It is a measure to identify how well multiple annotators can make the same annotation decision. Annotated agreements were asked to review by two people on whether they agree or disagree with the given annotations. The reviews from the raters are available in Github [52, 53].

Cohen's Kappa Calculation was conducted to find the IAA. Raters were asked to mark as '1' if they agree with the annotation otherwise they added '0'. Cohen's Kappa [54] score was identified as **0.87** for the annotations. This represents an almost perfect agreement between the raters.

5.2 Data description

Totally, 1604 valid documents were collected and available publicly in Github[6].

5.2.1 Language breakdown

Table 8 shows the language breakdown of the collected documents.

Table 8
Language breakdown in the collected data

Language	Count
English	724
Sinhala	465
Tamil	415

[5] https://github.com/VIthulan/multilingual-doc-clustering/blob/main/src/data_validator.py
[6] https://github.com/VIthulan/multilingual-doc-clustering/tree/main/src/data/valid_records

5.2.2 Categories breakdown

Fig.10 shows the categories breakdown of the collected data. Most of the documents belong to economic, general, crime and political categories.

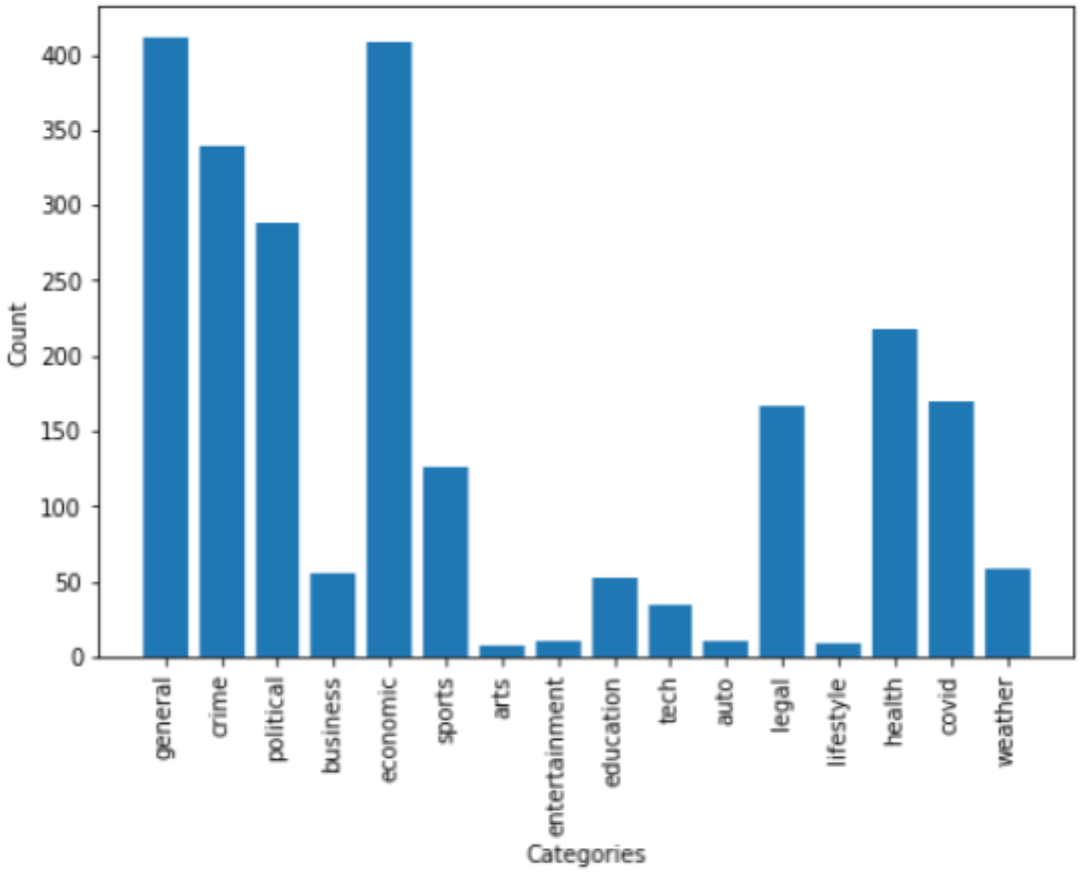


Fig.10. Categories breakdown in the collected data

5.3 Results

The TF-IDF model did not form multilingual clusters i.e., a cluster only contained documents from only one language. This could be probably due to missing shared vocabularies in the documents across the languages. TF-IDF does not solve this research problem. Therefore, baseline was obtained using mBERT and One-Pass clustering algorithm using both the Title and the Content of the document. It produced the accuracy value of 0.677.

Even-though mBERT does not include Sinhala, it was noted that the mBERT formed clusters with English, Sinhala and Tamil languages. Tables 9.1, 9.2 and 9.3 are a few examples of mBERT based clustered documents. Therefore, the mBERT baseline is comparable against the XLM-R performance.

Table 9.1
A correct cross-lingual cluster with Ta and En that was formed using mBERT

Content	Language	Context
“Suryakumar Yadav ruled out of T20I series vs Sri Lanka.”	En	Indian cricketer Suryakumar was ruled out of T20 Cricket match against Sri Lanka
“இந்திய அணி நாணய சுழற்சியில் வெற்றி”	Ta	The Indian cricket team has won the toss against Sri Lanka in the second T20 cricket match.
“அவுஸ்திரேலியா எதிர் இலங்கை: இறுதி T20 இன்று”	Ta	Sri Lanka is playing T20 cricket against Australia today

Table 9.2
A correct cross-lingual cluster with Ta, Si and En that was formed using mBERT

Content	Language	Context
“President directs to remove price disparity in liquid milk purchase.”	En	The Economic crisis and commodity prices go up.
“භාණ්ඩ මිල වැඩිවීම රාජ්‍ය ඇමති පිළිගනී”	Si	Minister of State expects increase in commodity prices
“நாளை(01) முதல் ஒரு கிலோ கிராம் மஞ்சள் 2,400 ரூபா”	Ta	Turmeric will cost Rs.2400 from tomorrow.

Table 9.3
A correct cross-lingual cluster with Si and En that was formed using mBERT

Content	Language	Context
“36 hour water cut in Colombo on Friday (18)”	En	Dry season and Water cut
“Daily power cuts from today?”	En	Shortage of hydro power due to the dry season. Shortage of Fuel. Economy crisis
“හෙට් පැය 5කට ආසන්න විදුලි කප්පාදුවක්”	Si	5 hours power cut. Shortage of Fuel. Economy crisis
“දෛනික ඉන්ධන අවශ්‍යතාව	Si	Daily demand for fuel increases.

ඉහළට”		Economy crisis

5.3.1 XLM-R Base and One-Pass clustering

Summary of the experiments can be seen in the Table 10.

Table 10.
XLM-R Base and One-Pass clustering experiments summary

XLM-R Base and One-Pass clustering experiments	**Accuracy**
Title and content	0.543
Title without text preprocessing	0.624
Title with text preprocessing	0.647
Content only with text preprocessing	0.642
Title and trimmed content with text preprocessing	0.694
Trimmed content with text preprocessing	**0.71**
Baseline (mBERT with Title and Content)	0.677

Title and content was experimented with the XLM-R Base model as well. XLM-R Base model produced 0.543 accuracy with multilingual clusters. It was noted that XLM-R Base has formed a lot of single document clusters at higher threshold values of the one-pass clustering algorithm.

Then only the title of the document was experimented with the XLM-R Base model. This method obtained the accuracy of 0.624 at the threshold 0.7. This approach has improved the accuracy compared to the accuracy of title and content in XLM-R Base.

Document title was pre-processed before the experiment. Text pre-processing has improved the accuracy by 2.3%. Hence, all of the following experiments were conducted on top of preprocessed text.

Only the content also experimented with XLM-R Base. This method obtained the accuracy of 0.642 at a very lower threshold 0.3. Content based clustering was forming a lot of single document clusters even at a smaller one-pass threshold values.

Since previous studies [32, 40] suggested that MLLM models performed better in smaller text documents, this research also conducted on trimmed content and trimmed content with title based experiments.

Trimmed content was able to produce better accuracy under content length less than 200. Beyond that it decreases the performance. Title and Trimmed content produced better results under 150 content length. Table 11 shows the accuracy comparison against the content length. As observed in the literature, XLM-R also performed better in smaller text lengths [32]. Similarly, in this research too XLM-R outperformed the mBERT baseline by 3.3% in shorter texts.

Table 11.
Accuracy comparison against the content lengths in XLM-R Base

Content length	**Accuracy of trimmed content**	**Accuracy of Title and trimmed content**
75	0.604	0.585
125	0.626	**0.695**
175	**0.71**	0.585
225	0.493	0.650

Though the performance is fluctuating with content length, one cannot clearly say the optimal content length for better performance. It depends on the content the document has in the selected first n characters.

5.3.1.1 XLM-R Base and One-Pass clustering summary

Best performance was obtained with a trimmed content approach. Also, it was observed that the XLM-R base performance increases as the document length decreases. Tabe 10 shows the summary of XLM-R Base and One-Pass clustering based experiments.

The performance of XLM-R has increased from 0.543 to 0.71. It has surpassed baseline in two approaches:

1. Trimmed content and Title
2. Trimmed content.

5.3.2 XLM-R Base and K-Means clustering

Since the number of classes is known prior as sixteen, K-Means clustering can also be experimented. Table 12 shows the summary of the results.

Table 12.
XLM-R base and K-Means clustering accuracy summary

XLM-R Base and K-Means experiments	**Accuracy**
Title only	0.305
Content only	0.507
Title and content	**0.536**
Trimmed content	0.308
Title and trimmed content	0.313
Baseline (mBERT)	**0.677**

It is visible that K-Means clustering does not perform better than the one-pass clustering algorithm with XLM-R Base nor the mBERT baseline. It can be due to the curse of dimensionality that comes with the XLM-R word embedding. In contrast to the one-pass algorithm, here K-Means provide comparatively better accuracy with longer text values.

5.3.3 XLM-R Large and One-Pass clustering

Table 13.
Summary of XLM-R Large based clustering approaches

XLM-R Large and One-Pass clustering experiments	**Accuracy**
Title and content	0.543
Title	0.683
Content only	0.642
Title and trimmed content	0.695
Trimmed content	**0.745**
Baseline	0.677

Similar to XLM-R Base experiments, Title only experimented with the XLM-R Large model as well. XLM-R Large with title has produced higher accuracy than the XLM-R Base title method (0.64). It also outperformed the mBERT baseline. XLM-R Large and content only approach did not improve from XLM-R Base or the mBERT.

XLM-R Large with trimmed content method obtained an accuracy of 0.745 at the threshold 0.775 and content length 75 characters. This accuracy has outperformed the XLM-R base model and also the mBERT by 6.8%.

XLM-R Large with title and trimmed content method obtained an accuracy of 0.695 at the threshold 0.6 and content length 125. This did not improve the accuracy of the XLM-R base model. Both produced similar results. This approach also improved the performance from the baseline.Table 14 below shows the accuracy comparison against the content lengths.

Table 14

Accuracy comparison against the content lengths in XLM-R Large

Content length	**Accuracy of trimmed content**	**Accuracy of Title and trimmed content**
50	0.710	0.668
75	**0.745**	0.684
100	0.734	0.612
125	0.722	**0.695**

5.3.3.1 XLM-R Large and One-Pass clustering summary

It was observed that XLM-R-Large performs better than XLM-R-Base version and sometimes, both the versions perform equally well. XLM-R-Large trimmed content approach outperformed both the baseline and the XLM-R-Base. Summary can be seen in the Table 13.

Following are a few sample documents from the clusters. Table 15 and 16 show few sample multilingual clusters with similar context were formed together.

Table 15.

A correct cross-lingual cluster with Ta and Si, that was formed using XLM-R Large

Content	**Language**	**Context**
“உயர்தரப் பரீட்சை எழுதும் பரீட்சார்த்திகளுக்கான்	Ta	Special announcement for students

விசேட அறிவித்தல்”		who are to take advanced level examinations. - Education
“ශිෂ්යත්ව උත්තර පත්ර ඇගයීම් කටයුතු අවසන් අදියරේ”	Si	Scholarship answer scripts are in the final stages of evaluation - Education
“සා/පෙළට අයදුම්පත් කැඳවීම අද්න අවසන්”	Si	Call for O/L application ends today - Education

Table 16

A correct cross-lingual cluster with Ta, Si and En, that was formed using XLM-R Large

Content	Language	Context
“இந்திய அணிக்கு 184 ஓட்டங்கள் வெற்றி இலக்கு”	Ta	Indian cricket team needs to get 184 to win - Sports.
“இந்திய அணி நாணய சுழற்சியில் வெற்றி”	Ta	Indian cricket team won the toss - Sports
“භානුක ඉල්ලා ක්රිකට් ආයතනය ඉදිරිපිට විරේධයක්”	Si	Sri Lankan Cricketer Bhanuka demands protest in front of Cricket Institute - Sports
“தொடரை கைப்பற்றியது இந்தியா!”	Ta	Indian cricket team won the tour - Sports
“SL tour of India to begin on 24 Feb. with T20I series”	En	Sports
“කොදෙවි ක්‍රීඩිකාව රැකගත් සුපිරි උඩපන්දුව”	Si	Super catch by West indies - Cricket, Sports

5.3.3.2 Language-level performance breakdown

Table 17

Language level accuracy breakdown in XLM-R Large

Approach (XLM-R Large)	En	Si	Ta
Title and content	**0.506**	0.274	0.446
Title	0.473	0.541	**0.631**
Content only	**0.642**	0.313	0.333

Title and trimmed content	**0.650**	0.557	0.548
Trimmed content	0.608	0.684	**0.691**

It can be seen in Table 17 that XLM-R Large provides better performance in longer text for English language and better performance in shorter text for Tamil and Sinhala languages.

5.4 Discussion

Table 18
Summary of the accuracy scores among different approaches

Comparing different word embedding and clustering techniques	**mBERT**	**XLM-R Base**	**XLM-R Base**	**XLM-R Large**
	One-Pass	*One-Pass*	*K-Means*	*One-Pass*
Title and Content	**0.677**	0.543	**0.536**	0.543
Title only	-	0.647	0.305	0.683
Content only	-	0.642	0.507	0.642
Title and trimmed content	-	0.694	0.313	0.694
Trimmed content	-	**0.71**	0.309	**0.745**

Summary of the accuracy results can be seen in Table 18.

It can be seen that XLM-R-Base outperformed mBERT in “Title and trimmed content” and “Trimmed content” based input variations. The XLM-R-Large model outperformed mBERT in following input variations:

- Title only,
- Title and trimmed content
- Trimmed content

XLM-R-Large model performed on par to XLM-R-Base in 2 occasions and outperformed XLM-R-Base model in other 3 occasions. Overall best performance of 0.745 was obtained by XLM-R Large with trimmed content approach. Fig.11 shows the XLM-R performance of comparison against the baseline mBERT model.

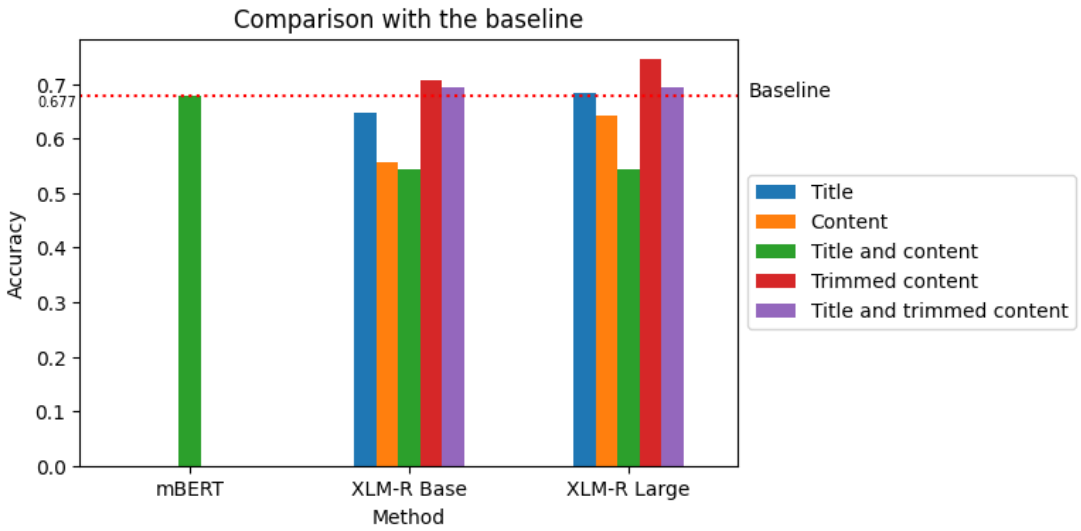


Fig.11 Performance comparison of XLM-R models against the baseline

Best performance with XLM-R Base was obtained with a trimmed content approach using 175 word tokens. Also, it was observed that the XLM-R base performance increases as the document length decreases.

It is visible that K-Means clustering does not perform better than the one-pass clustering algorithm. It can be due to the curse of dimensionality that comes with the XLM-R word embedding. In contrast to the one-pass algorithm, K-Means provides comparatively better accuracy with longer text values (Eg.Title and content, Content). Fig. 12 depicts the accuracy comparison against One-Pass and K-Means.

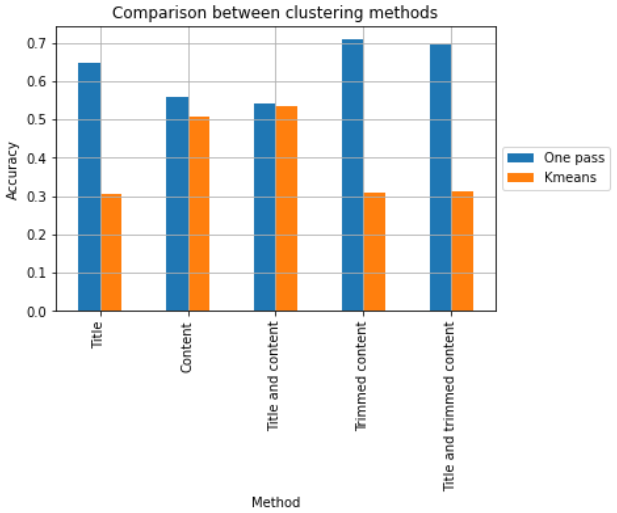


Fig.12 Performance comparison for KMeans and One Pass

It was observed that XLM-R Large performs better than XML-R Base version and sometimes, both the versions perform equally well. XLM-R Large trimmed content approach outperformed both the baseline and the XLM-R Base. Similar to XLM-R Base, XLM-R Large also performed better with shorter texts like title and trimmed contents. Fig.13 shows the comparison between XLM-R Large and XLM-R Base.

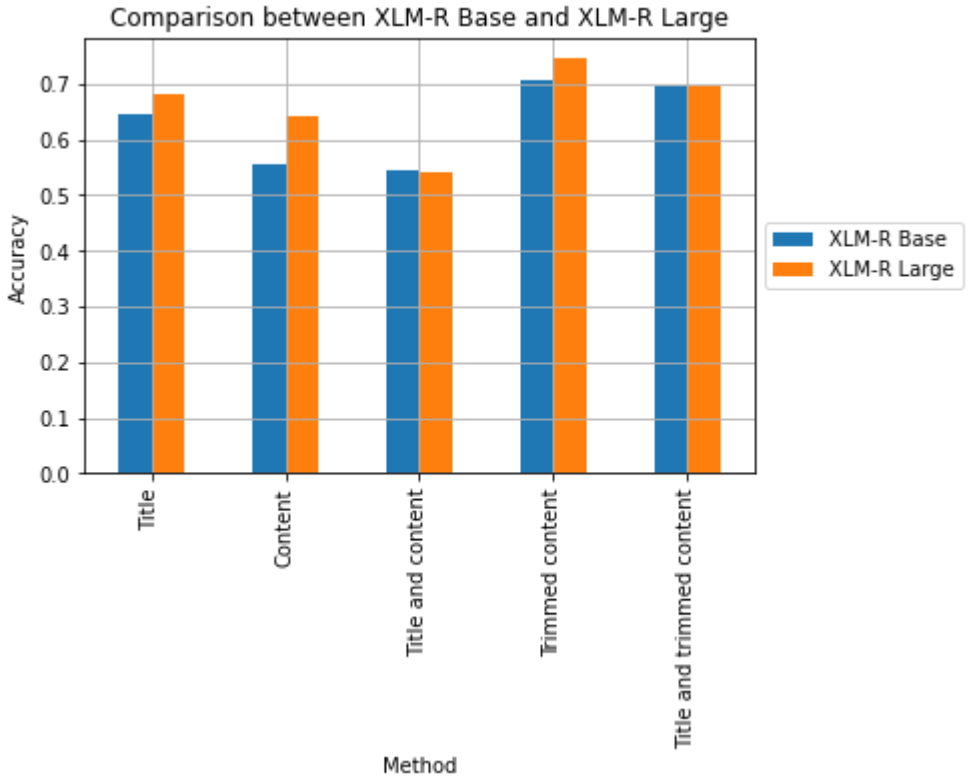


Fig.13 Performance comparison between XLM-R Base and XLM-R Large models

Table 19.
Language level performance breakdown with XLM-R Large and One-Pass

Approach	En	Si	Ta
Title and content	**0.506**	0.274	0.446
Title	0.473	0.541	**0.631**
Content only	**0.642**	0.313	0.333
Title and trimmed content	**0.650**	0.557	0.548
Trimmed content	0.608	0.684	**0.691**

It can be seen in Table 19 that XLM-R Large provides better performance in longer text (Content only, title and trimmed content) for English language and better performance in shorter text (title, trimmed content) for Tamil and Sinhala languages.

Park et al. [31] shows that traditional clustering algorithms cannot perform better with the MLLM generated word embeddings due to the high dimensionality. This justifies the lower performance of the K-Means clustering algorithm seen in this research. Few researchers [32, 40] observed that MLLM performed better with lower number of tokens, and similarly XLM-R performed better with smaller number of tokens in this research as well.

5.4.1 Error analysis

Following documents were grouped in a single cluster and the order of the documents shows the order of documents that the cluster was formed.

Table 20
Erroneous cluster formed due to order of the documents

Content	Language	Context
“police in sialkot have submitted a challan to the prosecution , naming 88 suspects in the killing”	En	Police, Crime
“a police constable has been killed after being assaulted by a group of individuals in tangalle.”	En	Police, Crime
“minister namal rajapaksa emphasises the need for sri lanka to ' go digital '”	En	Political
“health minister keheliya rambukwella has long - overdue electricity bill arrears for a colombo 7”	En	Political, Power shortage
“the electricity supply will be interrupted across the island in two slots today (february 18)”	En	Power shortage
“the haliela police launched investigations after a school girl had been axed to death at uduwara”	En	Police, Crime
“parts of the country may experience interruptions to the power supply from today”	En	Power shortage

As shown in the above Table 20, the cluster was formed with police and crime related documents. Then somehow a political document was added into the cluster. That caused the addition of another political document with power crisis related information. That made intaking a couple more power crisis related documents into the cluster. This reduced the quality of the cluster. One-Pass algorithm groups the documents iteratively in an order and the centroid is updated as the mean of all

documents. Correct and incorrect documents which are classified into the cluster have an equal weight in determining the centroid. Hence the order that the documents are clustered plays a major role in cluster quality.

Table 21
Erroneous cluster with shared word

Content	Language	Context
“சர்வதேச தாய்மொழி தினம் (international mother language day) பிரதமர் மஹிந்த ராஜபக்ஷ அவர்களின் தலைமையில..”	Ta	PM Mahinda Rajapakshe to attend **International** mother language day
“ஐக்கிய நாடுகளின் மனித உரிமைகள் பேரவையின் 49 ஆவது கூட்டத்தொடர் அடுத்த வாரம் ஜெனிவாவில் ஆரம்பமாகவுள்ளத..”	Ta	49th UN human rights council to commence in Geneva
“சிங்கப்பூரில் நடைபெறும் சர்வதேச் பளுதூக்கல் போட்டியில் 73 கிலோகிராம் எடைப் பிரிவில் இந்திக திசாநாயக்..””	Ta	Indika Dissanayaka is to participate in the **International** weight lifting competition under 73kg category

As shown in the above Table 21 three Tamil language documents were grouped into a cluster because of the term “**சர்வதேச**” (International), though, these documents don't share the same subject to be in the same cluster. Second document doesn’t explicitly say “international”, however, it was added into the document due to its context.

Table 22
Erroneous cluster with shared context

Content	Language	Context
“ரஷ்யா – உக்ரைனுக்கிடையிலான மோதல் இன்று இரண்டாவது நாளாகவும் தொடர்கின்றது . ரஷ்யாவின் நடவடிக்கைகளுக்கு..”	Ta	Ukraine-Russia war continues in 2nd day
“ත්‍රිකුණාමලය , කින්නියා ප්‍රදේශයේදී වෙඩි තැබීමක් සිදුකර තිබේ . අප වාර්තාකරු පැවසුවේ ඊයේ (07) රාත්‍රියෙ..”	Si	A shooting took place in the Kinniya area in Trincomalee.

Table 22 shows that the documents were grouped together based on “shooting”. However, the first document is about international war and politics whereas the latter one is a local crime.

6.0 CONCLUSION

Pre-trained MLLMs such as mBERT, XLM-R were used in this research to employ cross-lingual document clustering for news articles written in Sinhala, Tamil and English languages. 1604 annotated quality verified data has been collected in Sinhala, Tamil and English languages from 16 sources. The data is publicly available in the Github repository[7] [8] for anyone to continue the research with the Cross-lingual documents. Sixteen categories have been identified and the collected documents were annotated. Inter annotator agreement was calculated using Cohen's Kappa Statistic. This provided the statistical value of 0.87 which translates into almost perfect agreement. Research was conducted using mBERT as baseline, XLM-R base, XLM-R large for word embedding and, One-pass and K-Means for clustering. 5 different input variations were used. Baseline score was set as 0.677 with mBERT. XLM-R Base provided best results of 0.71 accuracy using the trimmed content and one-pass clustering algorithm. XLM-R Large provided best results of 0.745 accuracy using the trimmed content and one-pass clustering algorithm. Both have outperformed the mBERT based baseline (Table 13).

Even though mBERT does not include Sinhala, it was able to produce correct cross-lingual clusters. It shows the zero shot capability of the mBERT. K-Means clustering algorithm did not perform well and provided inferior results against one-pass based solution. It could be due to the curse of dimensionality. XLM-R Large provides better performance in longer text (Content only, title and trimmed content) for English language and better performance in shorter text (title, trimmed content) for Tamil and Sinhala languages. Performance of the model was impacted due to the order of documents that were clustered using the One-pass algorithm. It was noted that documents that contain similar words were added into the same cluster even though they don't share the same topic.

For the best of our knowledge, there are no other MLLMs that support Sinhala, Tamil and English. Therefore 0.745 can be set as the novel baseline for the cross-lingual document clustering for Tamil, Sinhala and English documents.

Research can be extended with classification methods since the classes are known. Knowledge distillation with teacher-student architecture can be implemented for the word embedding. Vector alignments can be experimented with XLM-R Base and XLM-R Large.

---

[7] https://github.com/VIthulan/multilingual-doc-clustering/tree/main/src/data/valid_records

[8] https://github.com/VIthulan/multilingual-doc-clustering/blob/main/src/processed_data/reviewed/documents_annotated_v2_1.csv

7.0 REFERENCE

[1] N. Y. Saiyad, H. B. Prajapati and V. K. Dabhi, "A survey of document clustering using semantic approach," 2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT), 2016, pp. 2555-2562, doi: 10.1109/ICEEOT.2016.7755154.

[2] Facebookresearch, "facebookresearch/XLM: PyTorch original implementation of Cross-lingual Language Model Pretraining.," GitHub. [Online]. Available: https://github.com/facebookresearch/XLM. [Accessed: 13-Sep-2021].

[3] M. F. Fayaza and S. Ranathunga, "Tamil News Clustering Using Word Embeddings," 2020 Moratuwa Engineering Research Conference (MERCon), 2020.

[4] P. Nanayakkara and S. Ranathunga, "Clustering Sinhala News Articles Using Corpus-Based Similarity Measures," 2018 Moratuwa Engineering Research Conference (MERCon), 2018.

[5] "A Fast and Powerful Scraping and Web Crawling Framework," Scrapy. [Online]. Available: https://scrapy.org/. [Accessed: 13-Sep-2021].

[6] Sparck Jones Karen, " A statistical interpretation of term specificity and its application in retrieval," Document Retrieval Systems, pp. 132–142, 1988.

[7] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, en J. Dean, "Distributed Representations of Words and Phrases and Their Compositionality", in Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2, Lake Tahoe, Nevada, 2013, bll 3111–3119.

[8] C. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. Bethard, and D. Mcclosky, "The Stanford CoreNLP Natural Language Processing Toolkit," Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations, 2014.

[9] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching Word Vectors with Subword Information," Transactions of the Association for Computational Linguistics, vol. 5, pp. 135–146, 2017.

[10] E. Grave, P. Bojanowski, P. Gupta, A. Joulin, en T. Mikolov, "Learning Word Vectors for 157 Languages", arXiv [cs.CL]. 2018.

[11] O. Melamud, J. Goldberger, en I. Dagan, "context2vec: Learning Generic Context Embedding with Bidirectional LSTM", in Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning, 2016, bll 51–61.

[12] B. McCann, J. Bradbury, C. Xiong, en R. Socher, "Learned in Translation: Contextualized Word Vectors", in Proceedings of the 31st International Conference on Neural Information Processing Systems, Long Beach, California, USA, 2017, bll 6297–6308.

[13] M. E. Peters et al., "Deep Contextualized Word Representations", in Proceedings of the 2018 Conference of the North American Chapter of the

Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), 2018, bll 2227–2237.
[14] J. Devlin, M.-W. Chang, K. Lee, en K. Toutanova, BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. 2018.
[15] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, en I. Sutskever, “Language Models are Unsupervised Multitask Learners”, 2018.
[16] Google-Research, “google-research/bert: TensorFlow code and pre-trained models for BERT,” GitHub. [Online]. Available: https://github.com/google-research/bert. [Accessed: 17-Sep-2021].
[17] C. Raffel et al., “Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer”, arXiv [cs.LG]. 2020.
[18] M. Müller, M. Salathé, en P. E. Kummervold, “COVID-Twitter-BERT: A Natural Language Processing Model to Analyse COVID-19 Content on Twitter”, arXiv [cs.CL]. 2020.
[19] U. Naseem, I. Razzak, S. K. Khan, en M. Prasad, “A Comprehensive Survey on Word Representation Models: From Classical to State-Of-The-Art Word Representation Language Models”, arXiv [cs.CL]. 2020.
[20] S. Doddapaneni, G. Ramesh, A. Kunchukuttan, P. Kumar, en M. M. Khapra, “A Primer on Pretrained Multilingual Language Models”, arXiv [cs.CL]. 2021.
[21] H. W. Chung, T. Févry, H. Tsai, M. Johnson, en S. Ruder, “Rethinking embedding coupling in pre-trained language models”, arXiv [cs.CL]. 2020.
[22] X. Ouyang et al., “ERNIE-M: Enhanced Multilingual Representation by Aligning Cross-lingual Semantics with Monolingual Corpora”, arXiv [cs.CL]. 2021.
[23] XTREME. [Online]. Available: https://sites.research.google/xtreme. [Accessed: 25-Sep-2021].
[24] J. Hu, S. Ruder, A. Siddhant, G. Neubig, O. Firat, en M. Johnson, “XTREME: A Massively Multilingual Multi-task Benchmark for Evaluating Cross-lingual Generalization”, arXiv [cs.CL]. 2020.
[25] T. Pires, E. Schlinger, en D. Garrette, “How multilingual is Multilingual BERT?”, arXiv [cs.CL]. 2019.
[26] G. Lample en A. Conneau, “Cross-lingual Language Model Pretraining”, arXiv [cs.CL]. 2019.
[27] A. Conneau et al., “Unsupervised Cross-lingual Representation Learning at Scale”, arXiv [cs.CL]. 2020.
[28] Z. Chi et al., “XLM-E: Cross-lingual Language Model Pre-training via ELECTRA”, arXiv [cs.CL]. 2021.
[29] Google-Research, “bert/multilingual.md at master · google-research/bert,” GitHub, 18-Oct-2019. [Online]. Available: https://github.com/google-research/bert/blob/master/multilingual.md. [Accessed: 13-Sep-2021].

[30] M. Marcińczuk, M. Gniewkowski, T. Walkowiak, en M. Będkowski, “Text Document Clustering: Wordnet vs. TF-IDF vs. Word Embeddings”, in Proceedings of the 11th Global Wordnet Conference, 2021, bll 207–214.
[31] J. Park, C. Park, J. Kim, M. Cho, en S. Park, “ADC: Advanced document clustering using contextualized representations”, Expert Systems with Applications, vol 137, bll 157–166, 2019.
[32] L. Stankevičius en M. Lukoševičius, “Testing pre-trained Transformer models for Lithuanian news clustering”, arXiv [cs.IR]. 2020.
[33] N. Reimers en I. Gurevych, “Making Monolingual Sentence Embeddings Multilingual using Knowledge Distillation”, in Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, 11 2020.
[34] “Multilingual-Models¶,” Multilingual-Models - Sentence-Transformers documentation. [Online]. Available: https://www.sbert.net/examples/training/multilingual/README.html#available-pre-trained-models. [Accessed: 29-Sep-2021].
[35] C. Vu, “A complete guide to transfer learning from English to other Languages using Sentence Embeddings...,” Medium, 22-May-2020. [Online]. Available: https://towardsdatascience.com/a-complete-guide-to-transfer-learning-from-english-to-other-languages-using-sentence-embeddings-8c427f8804a9. [Accessed: 29-Sep-2021].
[36] “The open parallel corpus,” OPUS. [Online]. Available: https://opus.nlpl.eu/. [Accessed: 29-Sep-2021].
[37] S. Dutta, “‘Alignment is All You Need’: Analyzing Cross-Lingual Document Similarity for Domain-Specific Applications”, in CLEOPATRA@WWW, 2021.
[38] M. Artetxe en H. Schwenk, “Massively Multilingual Sentence Embeddings for Zero-Shot Cross-Lingual Transfer and Beyond”, arXiv [cs.CL]. 2019.
[39] I. Gusev en I. Smurov, “Russian News Clustering and Headline Selection Shared Task”, arXiv [cs.CL]. 2021.
[40] V. A. S and S. E. Y, “Russian News Similarity Detection with SBERT: pre-training and fine-tuning,” Computational Linguistics and Intellectual Technologies: Proceedings of the International Conference “Dialogue 2021,” Jun. 2021.
[41] M. Linger en M. Hajaiej, “Batch Clustering for Multilingual News Streaming”, arXiv [cs.CL]. 2020.
[42] S. Miranda, A. Znotiņš, S. B. Cohen, en G. Barzdins, “Multilingual Clustering of Streaming News”, arXiv [cs.CL]. 2018.
[43] G. Leban, B. Fortuna and M. Grobelnik, "Using News Articles for Real-time Cross-Lingual Event Detection and Filtering," in Proc. of the NewsIR’16 Workshop at ECIR, Padua, Italy, 2016. pp. 33-38

[44] S. Wu, A. Conneau, H. Li, L. Zettlemoyer, en V. Stoyanov, “Emerging Cross-lingual Structure in Pretrained Language Models”, arXiv [cs.CL]. 2020.
[45] S. Wu en M. Dredze, “Do Explicit Alignments Robustly Improve Multilingual Encoders?”, arXiv [cs.CL]. 2020.
[46] V. Vithulan, "multilingual-doc-clustering/template.json at main · VIthulan/multilingual-doc-clustering", GitHub, 2022. [Online]. Available: https://github.com/VIthulan/multilingual-doc-clustering/blob/main/src/template.json. [Accessed: 29- Mar- 2022].
[47] V. Vithulan, "multilingual-doc-clustering/src/data/valid_records at main · VIthulan/multilingual-doc-clustering", GitHub, 2022. [Online]. Available: https://github.com/VIthulan/multilingual-doc-clustering/tree/main/src/data/valid_records. [Accessed: 29- Mar- 2022].
[48] "Interest in Different Types of News - Digital News Report 2013", Reuters Institute Digital News Report, 2022. [Online]. Available: https://www.digitalnewsreport.org/survey/2013/interest-in-different-types-of-news-2013/. [Accessed: 29- Mar- 2022].
[49] "What are the different types of news?", Quora, 2022. [Online]. Available: https://www.quora.com/What-are-the-different-types-of-news. [Accessed: 29- Mar- 2022].
[50] V. Vithulan, "multilingual-doc-clustering/documents_annotated_v2_1.csv at main · VIthulan/multilingual-doc-clustering", GitHub, 2022. [Online]. Available: https://github.com/VIthulan/multilingual-doc-clustering/blob/main/src/processed_data/reviewed/documents_annotated_v2_1.csv. [Accessed: 29- Mar- 2022].
[51] V. Vithulan, "multilingual-doc-clustering/data_validator.py at main · VIthulan/multilingual-doc-clustering", GitHub, 2022. [Online]. Available: https://github.com/VIthulan/multilingual-doc-clustering/blob/main/src/data_validator.py. [Accessed: 29- Mar- 2022].
[52] V. Vithulan, "multilingual-doc-clustering/review_iaa_calc_export_team_1.xlsx at main · VIthulan/multilingual-doc-clustering", GitHub, 2022. [Online]. Available: https://github.com/VIthulan/multilingual-doc-clustering/blob/main/src/IAA/review_iaa_calc_export_team_1.xlsx. [Accessed: 29- Mar- 2022].
[53] V. Vithulan, "multilingual-doc-clustering/review_iaa_calc_export_team_2.xlsx at main · VIthulan/multilingual-doc-clustering", GitHub, 2022. [Online]. Available: https://github.com/VIthulan/multilingual-doc-clustering/blob/main/src/IAA/review_iaa_calc_export_team_2.xlsx. [Accessed: 29- Mar- 2022].
[54] V. Vithulan, "multilingual-doc-clustering/Inter-Annotater-Agreement.ipynb at main · VIthulan/multilingual-doc-clustering", GitHub, 2022. [Online]. Available: https://github.com/VIthulan/multilingual-doc-clustering/blob/main/src/IAA/Inter-Annotater-Agreement.ipynb. [Accessed: 29- Mar- 2022].

[55] V. Vithulan, "Google Colaboratory", Colab.research.google.com, 2022. [Online]. Available: https://colab.research.google.com/drive/1qoGMjyiFmuC-ehWluOaxePKBOWV0FDex?usp=sharing. [Accessed: 29- Mar- 2022].
[56] V. Vithulan, "multilingual-doc-clustering/tf-idf-baseline.ipynb at main · VIthulan/multilingual-doc-clustering", GitHub, 2022. [Online]. Available: https://github.com/VIthulan/multilingual-doc-clustering/blob/main/src/Baseline/tf-idf-baseline.ipynb. [Accessed: 29- Mar- 2022].
[57] M. Lewis κ.ά., 'BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension'. arXiv, 2019.
[58] "Evaluation of clustering", Nlp.stanford.edu. [Online]. Available: https://nlp.stanford.edu/IR-book/html/htmledition/evaluation-of-clustering-1.html. [Accessed: 05- Jul- 2022].