International Journal of Machine Learning and Cybernetics
https://doi.org/10.1007/s13042-023-01795-9

ORIGINAL ARTICLE

TTL: transformer-based two-phase transfer learning for cross-lingual news event detection

Hansi Hettiarachchi[1] · Mariam Adedoyin-Olowe[1] · Jagdev Bhogal[1] · Mohamed Medhat Gaber[1]

Received: 7 June 2022 / Accepted: 31 January 2023
© The Author(s) 2023

Abstract

Today, we have access to a vast data amount, especially on the internet. Online news agencies play a vital role in this data generation, but most of their data is unstructured, requiring an enormous effort to extract important information. Thus, automated intelligent event detection mechanisms are invaluable to the community. In this research, we focus on identifying event details at the sentence and token levels from news articles, considering their fine granularity. Previous research has proposed various approaches ranging from traditional machine learning to deep learning, targeting event detection at these levels. Among these approaches, transformer-based approaches performed best, utilising transformers' transferability and context awareness, and achieved state-of-the-art results. However, they considered sentence and token level tasks as separate tasks even though their interconnections can be utilised for mutual task improvements. To fill this gap, we propose a novel learning strategy named *Two-phase Transfer Learning (TTL)* based on transformers, which allows the model to utilise the knowledge from a task at a particular data granularity for another task at different data granularity, and evaluate its performance in sentence and token level event detection. Also, we empirically evaluate how the event detection performance can be improved for different languages (high- and low-resource), involving monolingual and multilingual pre-trained transformers and language-based learning strategies along with the proposed learning strategy. Our findings mainly indicate the effectiveness of multilingual models in low-resource language event detection. Also, TTL can further improve model performance, depending on the involved tasks' learning order and their relatedness concerning final predictions.

Keywords Transformer · Two-phase transfer learning · Cross-lingual event detection · News media

1 Introduction

Nowadays, a huge amount of data is generated, especially on the internet, mainly by social media platforms and online news agencies [1, 2]. However, a vast majority of this data is unstructured and cannot be easily understood. Also, the high amount of generation makes it harder for human beings to analyse data and extract important information manually. Thus, automated intelligent mechanisms are crucial for effectively extracting the information available in data. Such mechanisms will be beneficial to a wide range of applications, including knowledge base construction, question answering and text summarising [2–4]. In this research, we target automatically detecting events from news media text to support knowledge base constructions. We specially focus on detecting events at sentence (event sentence identification) and token (event trigger and argument extraction) levels of news articles considering their fine-grained information coverage. Developing such an approach would be beneficial to multiple parties, such as governments, disaster management teams and social and political science communities, but it has been a challenge due to the diversity and nuance in events and high accuracy requirements [5].

Considering the importance of event detection, various approaches have been proposed by previous research ranging from traditional machine learning (ML) to deep learning (DL), as further described in Sect. 2. Overall, the earlier

✉ Hansi Hettiarachchi
hansi.hettiarachchi@mail.bcu.ac.uk

Mariam Adedoyin-Olowe
mariam.adedoyin-olowe@bcu.ac.uk

Jagdev Bhogal
jagdev.bhogal@bcu.ac.uk

Mohamed Medhat Gaber
mohamed.gaber@bcu.ac.uk

[1] School of Computing and Digital Technology, Birmingham City University, Birmingham B4 7XG, UK

Published online: 08 March 2023

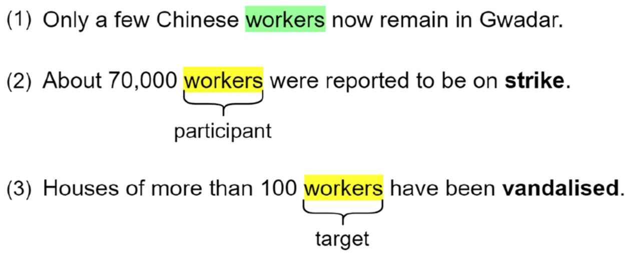

(1) Only a few Chinese workers now remain in Gwadar.
(2) About 70,000 workers were reported to be on **strike**.
participant
(3) Houses of more than 100 workers have been **vandalised**.
target

Fig. 1 Sample sentences from news articles with word *'workers'*. Bold text represents the triggers in event-described sentences. Word *'workers'* is highlighted in yellow if it represents an event argument and in green otherwise

work extensively relied on language-specific linguistic tools, resources and features, only focusing on high-resource languages such as English [6–8]. Such approaches mainly suffered from expandability issues and the inability to support low-resource languages. With the evolution of deep neural networks and their effectiveness, later research focused more on DL-based approaches to detect events [9–12]. This mostly eliminated the requirement to rely on linguistic tools, resources and features. However, deep networks require more instances for the training process, limiting their applicability when training data is scarce [13]. The other major challenge experienced by both traditional ML- and DL-based approaches is handling text ambiguity. For example, the word *'workers'* in the sentences in Fig. 1 plays three different roles. The first sentence does not describe any event, but the other two describe events expressed by the words (triggers) *'strike'* and *'vandalised'*. Thus, *'workers'* in sentence (1) is not event-related. However, *'workers'* in sentences (2) and (3) hold event arguments participant and target, respectively. It is crucial to focus on textual context to resolve such ambiguities while extracting event details.

Meta and transfer learning approaches have been popularly used in recent research to tackle data scarcity issues [14]. The main idea behind meta learning is learning to learn. It seeks an algorithmic solution for a problem with few training instances based on a set of models which perform a wide range of tasks [15]. Transfer learning pre-trains a model on an upstream dataset first and fine-tunes it on downstream tasks later, focusing more on learning representations and data source [14]. Due to the knowledge transfer, fine-tuning can be effectively done using a few training instances from the downstream task. Among these techniques, transfer learning has been popularly used recently [16, 17]. In the domain of natural language processing (NLP), this tendency is mainly influenced by the evolution of transformer-based language models or encoders (e.g. BERT), which can be pre-trained on the unlabelled text and fine-tuned for a wide range of downstream tasks [18]. Also, transformer architecture is capable of capturing contextual details in the text, disambiguating word senses. For simplicity, we will refer to the transformer encoder models as 'transformers' in the below content.

Transformer-based approaches have also been proposed for event detection recently, setting the state-of-the-art performance [5, 19]. However, to the best of our knowledge, all the available transformer-based approaches for news media event detection considered sentence and token level detection as two separate tasks and built separate models per task, ignoring their interconnections, which are helpful for mutual learning. Targeting this gap, in this research, we propose a novel transfer learning strategy named *Two-phase Transfer Learning (TTL)* based on transformers. This strategy allows the model to learn a task, following another related task in different data granularity (i.e. sentence or token), transferring the knowledge from the first task. We apply this learning for sentence and token level event detection tasks involving different pre-trained transformer models and comprehensively discuss their performance and involved tasks' transferability in this paper.

We also investigate how pre-trained transformer models can be effectively used in cross-lingual event detection, reporting a comprehensive experimental study, which was not available with previous work as far as we are aware. We use the multilingual version of GLOCON gold standard dataset [4], which has sentence and token level data, covering three languages: English, Portuguese and Spanish, for our experiments. At the sentence level, the training data distribution over these languages is approximately 23:1:3. At the token level, the English training dataset is 37 times larger than other language datasets. These statistics mainly explain the wide usage and data availability of English. Based on them, we consider English as a high-resource language and the other two as low-resource languages. We involve different language-based learning strategies: monolingual, multilingual, transfer and zero-shot learning, and different pre-trained transformer models for our experiments to analyse their impact on high- and low-resource language predictions at the sentence and token levels of event detection. We further extend these analyses with TTL to investigate its performance with other language-based learning strategies. Figure 2 illustrates a summary of the learning strategies we devised in this study, including the explored applications using different data types. To maintain simplicity, we did not include zero-shot learning in this diagram because it is applicable to all other strategies.

In summary, the main contributions of the paper are as follows.

1. We propose a novel learning strategy named *Two-phase Transfer Learning (TTL)*, involving different levels of data granularity and the capabilities of state-of-the-art transformer models, and release its implementation as

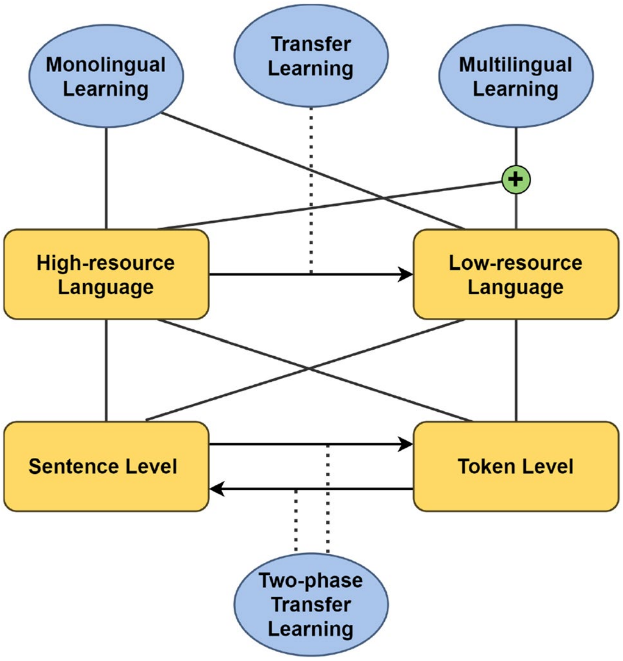


Fig. 2 Learning strategies involved in this study

an open-source project[1] to support related research and applications.

2. We apply the proposed strategy to sentence and token level tasks of news media event detection and discuss its effectiveness and applicability.
3. We empirically evaluate how the performance of news media event detection at the sentence and token levels can be improved for low-resource languages involving language-based learning strategies and cross-linguality in transformer models along with TTL, answering the following research questions:

RQ1: Can an event detection model based on a multilingual transformer, which is only fine-tuned for a particular language, outperform a model based on a monolingual transformer of that language?

RQ2: Can a high-resource language improve the event detection performance of a low-resource language using the cross-linguality in transformer models?

RQ3: Can two-phase transfer learning on transformers using different event detection tasks improve the performance of involved tasks in monolingual and multilingual settings?

The rest of this paper is organised as follows. Section 2 discusses the previous work on event detection in news media, covering sentence and token level tasks. Section 3 details the problem targeted by this research. Section 4 introduces transformer-based neural network architectures for event detection, the proposed learning strategy (TTL) and the involved language-based learning techniques. Section 5 describes the experimental setups we used for our experiments, including the datasets, pre-trained transformers and evaluation metrics. Section 6 comprehensively describes the conducted experiments and obtained results along with discussions which address the targeted research questions. Finally, Sect. 7 summarises the conclusions with aimed future work.

2 Related work

This section outlines the different approaches involved in previous research for sentence and token level event detection from news media text. Sentence level task targets recognising sentences that describe events, and token level task targets extracting event triggers and arguments from the event-described sentences. Overall, there was a high focus on supervised approaches to extract events from news media, mainly due to the less dynamicity of this media. Thus, we targeted supervised approaches proposed for sentence and token level extractions in our review. Also, we aimed to review recent papers, mainly published within the last decade, to maintain the recency of this review.

2.1 Event sentence identification

Event sentence identification is mostly considered as a sentence/text classification task. Previous research has proposed various approaches for this task, ranging from traditional machine learning (ML) to deep learning (DL). Recently, more focus has been given to DL-based methods, especially transformer-based models considering their effectiveness. More details of different approaches and their evolution over time are further discussed below.

Traditional machine learning: Early research commonly used text feature-based approaches with traditional classification algorithms to identify event sentences. For instance, [6] proposed using a Support Vector Machine (SVM) model trained on a wide range of features, including stemmed terms, part of speech (POS) tags, noun chunks, sentence length, sentence position and presence/absence of negative terms. Another research also utilised an SVM model to make predictions in Dutch text using different Bag of Word (BoW) features with token n-grams, character n-grams, lemma and POS tags, and special indicators such as numerals, symbols and time [20]. Similarly, the Logistic Regression algorithm was also involved in classifying event sentences using informative character n-gram and token unigram features

[1] Our implementation is publicly available on https://github.com/HHansi/MultiEventMiner.

[21]. However, as a major limitation, BoW ignores the word semantics and order, losing important information [10]. Even though n-grams capture word order to a certain extent, they lead to data sparsity issues [22]. Also, the involvement of language-based lexical features makes these approaches less expandable to different languages. Considering these limitations and following the effectiveness of word embedding models, deep learning-based approaches became more famous for text classification tasks in later research.

Deep learning: Among different neural networks, Long Short-Term Memory (LSTM) [23] and Convolutional Neural Network (CNN) [24] models were popularly used for text classification by previous research. LSTMs can learn long-term dependencies using their memory cells more effectively than vanilla Recurrent Neural Networks (RNN) [22]. CNNs consist of multiple convolutional and pooling layers, which can capture local text features such as syntax and semantics of words within a sentence [9]. Mostly, word embeddings were used to input text to these networks. For instance, [10] used pre-trained Word2vec embeddings with an LSTM network to classify sentences. The same approach is followed by [12], but they used a modified network with an attention layer on top of LSTM layers. Another research proposed a joint CNN and LSTM network combining their characteristics and used Word2vec embeddings for the input layer [22]. More modified networks such as Convolutional RNN (CRNN), which stacks a convolutional layer on top of an RNN and CNN with Attention (CNNA) which has an attention layer on top of a CNN also suggested by previous work [25]. However, one major limitation of deep neural networks is the high labelled data requirements to effectively fine-tune model weights from scratch. Also, the traditional word embeddings do not capture contextual details in the text, which are essential to understanding sentences. Transformer-based approaches were proposed recently to overcome these limitations.

Transformers: Transformers were designed with the ability to fine-tune for a downstream task by transferring the knowledge gained during the pre-training process [18]. This knowledge transfer allows learning the downstream task effectively even with fewer training instances, overcoming a major limitation in deep neural networks. Also, the transformer architecture can preserve contextual details in the text while generating representations. Overall, transformers recently improved the performance of many NLP applications with state-of-the-art results [18]. Following this trend, transformers are also involved in event sentence identification. A simple linear layer is commonly added on top of the transformer model to support text classification. Following this approach, [26] used pre-trained monolingual and multilingual BERT [18] models to classify event sentences. Similarly, [27] used RoBERTa [28] English model. Rather than using a multilingual model, they suggest translating text in other languages to English to make predictions using their model. Also, XLM-R [29] model is commonly used for multilingual predictions [19, 30]. It generates cross-lingual embeddings, which attempt to ensure words with the same meaning in different languages map to almost the same vector. Thus, it showed improved results than other multilingual models and translation-based approaches, which could suffer from language errors. Deviating from the common approach, [31] suggested adding an LSTM layer on top of a transformer and getting soft voting of BERT, RoBERTa and DistilBERT as the final prediction. Also, another research experimented with the weighted ensemble of RoBERTa model and Lex-Stem: a two-channel CNN with normal and stemmed text [32]. Overall, these modified approaches did not outperform the simple architecture with a large pre-trained transformer and linear output layer, which can consider state-of-the-art for event sentence identification [33].

2.2 Event trigger and argument extraction

Event trigger and argument extraction is commonly considered as a token classification problem by previous research. Similar to event sentence identification, various approaches based on traditional machine learning (ML) and deep learning (DL) have been used in previous research for this extraction task. A trend to involve transformers is also noticed in recent research. We discuss more details about available approaches below.

Traditional machine learning: Most early works used linguistic features with classification models to extract event triggers and arguments. For example, [8] built separate classification models using the SVM algorithm, treating trigger and argument extraction as separate tasks. They used various linguistic features, including tokens, POS tags, dependency paths, and synonyms from semantic dictionaries, for their models. Another research proposed using cross-entity inference for event extraction, focusing on the possibility of missing events by only using the local features [7]. In addition to using the knowledge in the training corpus, they used information from the Web to understand the background of entities. They also involved SVM classifiers in making final predictions. Rather than treating trigger and argument extraction as separate tasks, [34] suggested a joint system based on structured perceptron with beam search, allowing to improve the predictions of each task mutually. This approach also highly depends on linguistic features such as POS tags, lemmas, synonyms and dependencies. Overall, following the complexities in event extraction, traditional approaches extensively rely on linguistic features or knowledge bases resulting in less generality across different languages. Thus, similar to the trend with event sentence identification, there was more focus on deep learning-based approaches afterwards, considering their ability to extract underlying features in text automatically.

Deep learning: With event token extraction also, LSTM and CNN are the most commonly used neural network architectures by previous research. However, Bidirectional LSTM (Bi-LSTM) models were used over LSTM since both past and future states of the sequence are important for token labelling. Also, rather than using simple linear layers, Conditional Random Fields (CRFs) were used for output generation since they take context into account. For instance, [11] used a Bi-LSTM network with a CRF layer to extract event entities. They incorporated Word2vec and GloVe embeddings to feed text into the network. Ref. [35] used the same architecture with fastText and Multilingual Unsupervised and Supervised Embeddings (MUSE) to extract triggers. Also, more advanced embeddings such as ELMo, character and POS were used with this architecture [21]. Different variants of CNNs were also proposed for event extraction. Ref. [9] involved separate Dynamic Multi-pooling CNNs (DMCNNs) with Word2vec embeddings to extract triggers and arguments. Another research used path-aware graph CN with BERT embeddings [36]. Like traditional approaches, some DL-based approaches also treated trigger and argument extraction as a joint task to allow mutual learning and mitigate error propagation. For instance, [37] trained a joint Bi-LSTM model with Word2vec embeddings. Ref. [3] added dependency bridges over Bi-LSTM to utilise dependency relations for joint learning. A combination of Bi-LSTM and DMCNN was also proposed using an advanced embedding layer formed by concatenating BERT, GloVe, entity type, POS and dependency relation embeddings for joint event extraction [2]. Overall, there was a high focus on improving network architectures to improve event extraction in previous research. However, similar to the scenario with event sentence identification, these networks require a large amount of data for the from-scratch learning limiting their usability and performance. Thus, there is a recent trend to use transformers, mainly considering their effectiveness and transferability.

Transformers: Transformers have been involved with event trigger and argument extraction recently, considering their effectiveness. Following the DL-based approaches' trends, [35] designed a network with a CRF layer on the BERT model to extract event triggers. They used monolingual and multilingual BERT models to analyse the performance in different languages. Following the simple approach, [38] added linear layers on the BERT model per token/word to extract triggers. They used a separate BERT-based model to extract arguments and occupied its input with the identified triggers following a pipelined approach. However, there was a comparatively high tendency to build joint models for trigger and argument extraction using transformers, considering their computational complexities, the interconnections of these tasks and error-propagation in pipelined approaches. Ref. [32] proposed a joint model by adding a Bi-LSTM and CRF layer on the RoBERTa model for event extraction. Targeting multiple languages, the XLM-R model was used with linear output layers per token, following the same trend noticed with event sentence identification [19, 39]. Also, this simple architecture outperformed other modifications, being the state-of-the-art for event trigger and argument extraction [33].

2.3 Summary

In summary, we can mention that transformer-based models have state-of-the-art results for event sentence identification and trigger and argument extraction, outperforming traditional ML- and DL-based approaches. However, as described above, most approaches treated these tasks separately without considering their interconnections. Being an exception, [21] proposed a bottom-up approach from token to sentence level. They used a BERT sequence labelling model with linear output layers to extract triggers and arguments and then labelled a sentence as an event sentence if it contains a trigger. This approach mainly suffers from error propagation and also does not account for the possibility of involving sentence level knowledge for token level predictions. Targeting these gaps, in this research, we aim to propose a novel transfer learning strategy with transformers, which can learn from sentence to token level and vice versa to utilise knowledge from one level to support the predictions at the other level.

Considering the language coverage of available methods, early research mostly focused only on English. However, there is an increased focus on different languages with transformers, mainly involving multilingual models. A few approaches also used translation-based techniques to support different languages, but multilingual models can be considered more effective since translations could suffer from language errors. However, to the best of our knowledge, there is no comprehensive study that analyses different transformer models' performance involving different learning strategies targeting multilingual event sentence, trigger and argument extraction available in the literature. Filling this gap, we target conducting a thorough analysis in this research using the commonly used learning strategies and the one we propose.

3 Problem definition

The problem targeted by this research is automatically detecting events in news articles. Different data granularities are targeted by previous research with event detection. At the coarsest level, news articles that contain interesting events are filtered [40]. Narrow downing the output, some approaches are focused on identifying events at the sentence level of news articles [12, 20]. Going for a further fine-grained level, extracting event details at the token level of sentences also targeted [37, 38]. Among these levels, we focus on detecting event details at the sentence and token levels in this research considering their fine granularity to extract more focused or detailed information. Also, we aim to preserve multilingualism in our approaches, including the ability to process low-resource languages.

Previous research used different definitions for events. For example, in [41], an event is considered as something that happens at a particular time and place. Automatic Content Extraction (ACE) Program[2] defined an event as a specific occurrence involving participants or something that happens or a change of state. Considering the available definitions, we generally define an event using the Definition 1.

Definition 1 *Event:* An incident or activity which happened at a certain time and was reported in a data source.

Rather than focusing on general events from different domains, mostly, previous research focused on specific events such as natural disasters [40], economic events [20] and political events [33]. Specific focus allows the algorithm to learn the characteristics of the targeted domain and make effective predictions. Also, in reality, users mostly need to know events in interesting domains more accurately rather than knowing all the events from different domains [1]. Following this tendency and requirement, we also focus on extracting specific event details in this research.

At the sentence level, we target recognising whether a sentence is an event sentence or not. From the computing perspective, this is a sequence classification problem with binary labels. Following ACE and Global Contentious Politics Dataset (GLOCON)[3] annotation manuals, we define an event sentence more comprehensively using the Definition 2.

Definition 2 *Event sentence:* A sentence that describes an event or contains an expression (word or phrase) directly refers to an event.

At the token level, we target extracting event triggers and arguments from sentences. Previous research treated event trigger and argument extraction as separate [9, 38] as well as joint [19, 42] tasks. Considering the recent applications and resource limitations, we aim to build a joint system in this research. Similar to the sentence level, from the computing perspective, this task is a token classification problem with multiple labels. We define an event trigger and argument using Definitions 3 and 4, following ACE and GLOCON manuals.

Definition 3 *Event trigger:* The main word that most clearly expresses an event occurrence.

Definition 4 *Event argument:* An entity, temporal expression, or value serves as a participant or attribute of an event.

In summary, this research aims to develop approaches for event sentence identification and event trigger and argument extraction with the ability to support different languages, including low-resource languages.

4 Methodology

This section presents our methodology for news media event detection at the sentence and token levels following the recent trends in natural language processing (NLP), specifically the successful applications of transformer-based models and their cross-lingual and knowledge transferring abilities. Section 4.1 describes the transformer-based neural network architectures we used for sentence and token level tasks. Following it, in Sect. 4.2, we propose a novel *Two-phase Transfer Learning (TTL)* strategy combining the characteristics of traditional transfer learning, multi-task learning and transformers. Using this approach, we aim to transfer knowledge from data at different granularities (i.e. sentence and token levels) in this research. Also, to the best of our knowledge, this is the first attempt to transfer knowledge from different data granularities to identify events in news text. Finally, Sect. 4.3 summarises the different language-based learning strategies we involved to analyse the cross-lingual capabilities of the proposed architectures.

4.1 Neural network architectures

We use transformer-based architectures for news media event detection following their success in various NLP tasks being the state-of-the-art [18, 43, 44]. Apart from providing strong results than Recurrent Neural Network (RNN)-based architectures, most transformers such as BERT [18], XLM-R [29] provide pre-trained language models on large corpora to support effective fine-tuning of downstream tasks. These models are composed of multi-layer bidirectional transformer encoders using the self-attention mechanism [45] to generate linguistically powerful contextual language representations. Such an encoder takes a text sequence as the input and returns sequence and token representations/embeddings, which can use to learn downstream tasks while preserving the linguistical features of the original text.

[2] Details of ACE are available on https://www.ldc.upenn.edu/collaborations/past-projects/ace.

[3] Details of GLOCON are available on https://glocon.ku.edu.tr/.

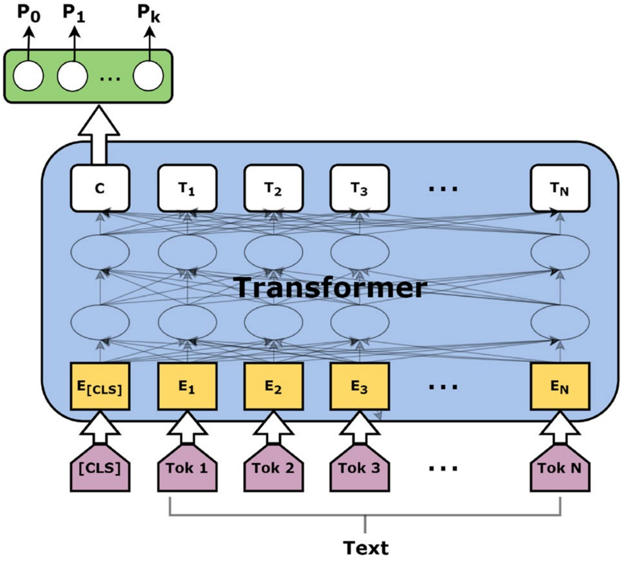


Fig. 3 Transformer-based sentence/sequence classification architecture

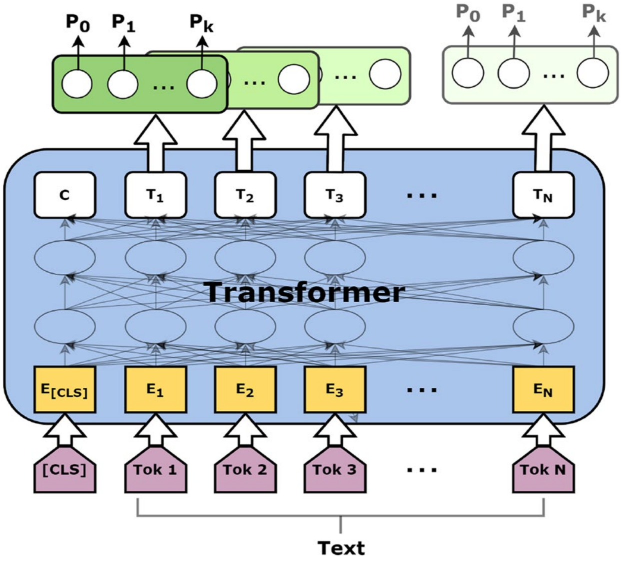


Fig. 4 Transformer-based token classification architecture

Transformer input format: Allowing to handle various downstream tasks, transformers are designed to take a single text sequence or a pair of sequences as the input. Different special tokens such as [CLS] and [SEP] are used to indicate the input text's organisation. [CLS] is added as the first token. If there are two sequences in the input, [SEP] is placed in between to indicate the separation. Following the raw text formatting, the text needs to convert to a token embedding using a tokeniser. Additionally, a segment embedding that holds boolean values (0 and 1), separating the segments and a position embedding with increasing numbers from 0, indicating the token positions are required to populate the final input. The sum of these three embeddings forms the input to a transformer model.

Transformer output format: The final hidden state of a transformer encoder provides representations for each token in the input. The first token ([CLS])'s output holds a representation corresponding to the entire sequence, which can be used as a contextual sequence embedding or with sequence-based predictions. The other outputs contain token representations per input token, which can be used as contextual word embeddings or with token-based predictions. To use a transformer model for a downstream task, an additional layer appropriate to the targeted task, like a classification head, needs to be put on top of the output layer.

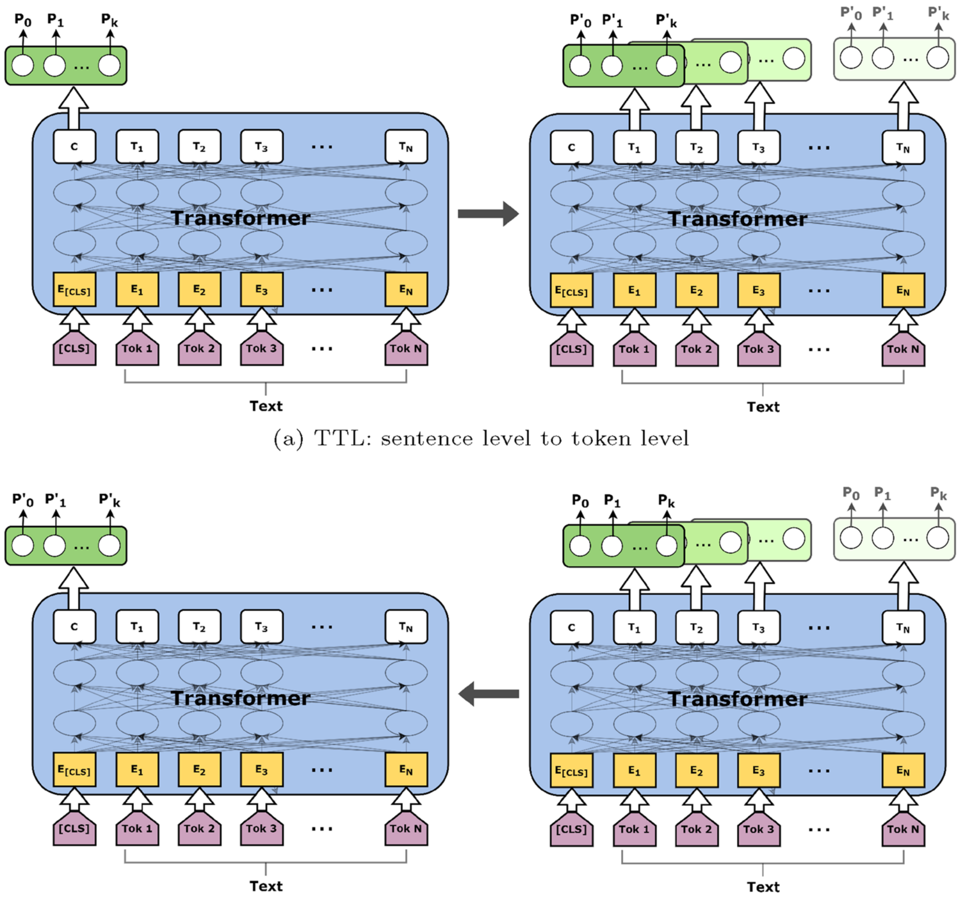
In this research, we target identifying event sentences and their triggers and arguments. We consider event sentence identification as a sequence classification problem and event trigger and argument extraction as a token classification problem. Both problems require processing a single sentence per instance. Thus, we only use the [CLS] token while formatting the inputs to the transformer without the [SEP] token. We add softmax layer(s) on top of the transformer model to conduct both classifications. For sequence classification, we feed the output of [CLS] to a softmax layer using the architecture shown in Fig. 3 since this output represents the entire sequence. For token classification, we feed the outputs of each token to separate softmax layers, as shown in Fig. 4. A softmax layer contains k neurons equivalent to the number of classes targeted by the classifier. Each neuron follows the softmax activation function in Eq. (1) returning probabilities per class (P_i). z_i and z_j represent input and output vectors. After calculating the probabilities per class, we pick the class with maximum probability as the final prediction of both tasks.

$$P_i = \frac{e^{z_i}}{\sum_{j=1}^{k} e^{z_j}} \tag{1}$$

4.2 Two-phase transfer learning (TTL)

Transfer learning (TL) is the process of improving a target predictive function of task T_t at a target domain D_t using the related knowledge gained from a task T_s at a source domain D_s where $D_s \neq D_t$ or $T_s \neq T_t$ [46]. This knowledge transfer also helps mitigate overfitting and underfitting problems that arise with deep neural networks due to data limitations, allowing to use such network capabilities for a wide range of tasks where training data is scarce [43, 47]. Mainly, there are two TL types based on the consistency between the source and the target feature and label spaces [48]. If both source and target feature and label spaces are equivalent ($X_s = X_t$ and $Y_s = Y_t$), it is named homogeneous TL, and if either

Fig. 5 Two-phase classification architecture



(a) TTL: sentence level to token level

(b) TTL: token level to sentence level

feature spaces or label spaces are not equivalent ($X_s \neq X_t$ and/or $Y_s \neq Y_t$), it is named heterogeneous TL. Comparatively, homogeneous learning is commonly used in previous research, but heterogeneous learning is more advantageous considering its ability to learn from different feature/label spaces [49]. However, most available solutions handle the heterogeneity by transforming feature/label spaces into common spaces with the possibility of losing important information in data or original data structure [50, 51].

The concept of multi-task learning (MTL) is popularly used in recent research to handle heterogeneous tasks [52, 53]. MTL optimises a model for more than one task simultaneously leveraging the generalisation across all tasks [54]. MTL learns the interconnections between tasks rather than transferring knowledge from a related task as with TL. Also, this learning does not require space transformations similar to heterogeneous TL. However, this strategy requires having shared training instances across all tasks, which are unavailable in many scenarios, including low-resource language-based predictions.

Considering the above limitations in heterogeneous TL and MTL, we propose a hybrid strategy named *Two-phase Transfer Learning (TTL)* in this research. We mainly utilise the characteristics of transformers for our approach. Transformer models are originally designed with the ability to fine-tune a pre-trained language model for a downstream task by adding an additional output layer [18]. This allows transferring the knowledge from the language model to the downstream task predictions. Following this idea, we propose fine-tuning a pre-trained transformer for two related tasks in two sequential phases, unlike the simultaneous learning that happens with MTL. We add different output layers to the model depending on the targeted task at each phase but share the transformer weights among the tasks allowing the phase-2 task to learn from the phase-1 task in addition to the original language model.

We target event detection tasks in two data granularities (i.e. sentence and token level) with TTL in this research, mainly to analyse how their relationships and data sizes affect the learning. These levels have intermediate relationships, specifically from the fine-grained (token) level to the coarse-grained (sentence) level, which helps derive the final labels. For example, if a sentence has an event trigger, it is an event sentence. Considering the data sizes, there is a tendency to have more labelled data at the sentence level than the token level due to the data annotation complexities at token data [4]. We use the transformer architectures introduced in Sect. 4.1 for sentence and token level classifications with TTL as shown in Fig. 5.

For the sentence to token level transfer, the transformer model is initially fine-tuned for the sentence level predictions by feeding the output of [CLS] to a softmax layer, which predicts probabilities per class in the sentence level, $P_0, P_1, ...P_k$, as illustrated in Fig. 5a. Then, the fine-tuned transformer weights are again fine-tuned for the token level predictions by feeding the output of each token to separate softmax layers, which predicts the token level class probabilities, $P^{'}_0, P^{'}_1, ...P^{'}_k$, utilising the transformer's pre-trained and phase-1 fine-tuned/sentence level knowledge. The same architectures are trained conversely for the token to sentence level transfer as shown in Fig. 5b. Initially, the transformer model is fine-tuned for the token level predictions by adding multiple softmax layers per token and then fine-tuned again for the sentence level predictions using a single softmax layer over the [CLS] output, transferring the transformer's pre-trained and phase-1 fine-tuned/token level knowledge for sentence level predictions.

4.3 Language-based learning

To analyse the cross-lingual capabilities of the proposed architectures, we involve the following language-based learning strategies for fine-tuning. These strategies are used in different areas, including event detection [5, 19], translation quality estimation [43] and word sense disambiguation [55], but to the best of knowledge, no comparison covering all the strategies for news media event detection is available. Furthermore, we analyse the impact of these strategies on TTL in this paper. We specifically focus on improving low-resource language predictions using the knowledge in high-resource language data.

1. *Monolingual learning* trains a model using data from a single language. This is the common learning strategy, and it mostly performs well for high-resource languages with the provision of enough data to fine-tune a transformer [5, 19].
2. *Multilingual learning* trains a model in multiple languages simultaneously. This strategy can supply more training data to the model, overcoming data scarcity in low-resource languages [19]. Also, multilingual learning can generally help optimise the model effectively for different languages capturing their interconnections, unlike monolingual learning. Additionally, a model that supports multiple languages is more resource-effective and easily manageable than a monolingual model collection. However, this learning is only applicable to multilingual transformers.
3. *Language-based zero-shot learning* uses a model fine-tuned for the same task in another language(s) to make predictions. It is commonly used when no training data are available for a particular language and is especially beneficial for low-resource languages [5, 55]. This strategy became more popular in NLP tasks recently following the cross-lingual abilities in transformer models.
4. *Language-based transfer learning* is a variant of TL that transfers knowledge from one language to another. This strategy fine-tunes a model learned in a particular language for the same task in another language. Popularly, models trained on high-resource languages are fine-tuned for low-resource languages following this idea [43, 56].

5 Experimental setup

This section presents the experimental setup of our architectures for event sentence identification and event trigger and argument extraction. We used a multilingual news event dataset, which is further described in Sect. 5.1 for our experiments. More details about the evaluation metrics we used are available in Sect. 5.2. Considering the targeted languages, we involved four popular transformer models, including a multilingual model, for our experiments, and their details are summarised in Sect. 5.3. The hyper-parameter configurations we used for our experiments are available in Sect. 5.4. Additionally, we follow a common convention to format training data combinations along with learning strategies while reporting results, and it is explained in Sect. 5.5. We implemented all the neural network architectures in Python 3.7[4] using the FARM library.[5] All our experiments are conducted on a GeForce RTX 3090 GPU.

5.1 Dataset

We use the multilingual version of GLOCON gold standard dataset [4] which is released along with the workshop on Challenges and Applications of Automated Extraction of Socio-political Events from Text (CASE) in 2021 [33] considering its recency, open-availability and coverage. This dataset is created targeting socio-political events covering demonstrations, industrial actions, group clashes, political violence, armed militancy and electoral mobilizations. It has data from three languages: English, Portuguese and Spanish, at different levels of granularity. Also, multiple news sources were used to collect data.

In this research, we target identifying event sentences and their trigger and argument spans. Thus, we only use

[4] Our codebase in publicly available on https://github.com/HHansi/MultiEventMiner.

[5] FARM is available on https://github.com/deepset-ai/FARM.

Table 1 Number of sentences in sentence and token level datasets

Language	Sentence level		Token level	
	Train	Test	Train	Test
English (En)	22481	1290	3248	311
Portuguese (Pt)	1001	1445	87	192
Spanish (Es)	2613	686	87	190

Table 2 Label distribution of token level data

Label	Number of spans		
	En	Pt	Es
Trigger	4595	122	127
Participant	2663	73	79
Place	1570	61	14
Target	1470	32	52
Organizer	1261	19	23
Etime	1209	41	32
Fname	1201	48	39

sentence and token level data of the GLOCON dataset for our experiments. Analysing the original data, we noticed some instances shared among training and testing splits at different levels. Since such occurrences could affect the performance of TTL, we removed those instances from the training splits. For example, if an instance in token level test data is available in sentence level training data, it was removed from the training split. Also, we removed URLs and repeating symbols from the sentence level data because they are uninformative. However, we did not apply any processing for token level data that had already been cleaned. The sizes of cleaned datasets are summarised in Table 1. Comparatively, English has more instances of being a high-resource language than others that can be considered low-resource languages. Considering the levels of granularity, the token level has less data than the sentence level due to the complexities associated with data annotation.

Sentence level data have binary labels indicating whether a sentence describes an event or not. Positive sample ratios for English, Portuguese and Spanish are 18%, 10% and 12%, respectively. There are many non-event sentences because full documents were sampled to get sentences without applying any filtering. Since this imbalance illustrates the real scenario and provides more training samples from the targeted domain to the models, we directly experimented with these data without pruning them. Token level data are provided with labels indicating event triggers and arguments. Overall, there are six argument types, and the details of their distributions over different languages are given in Table 2.

5.2 Evaluation metrics

Each architecture we experiment with has its own goal, following the *SOTA* approach [57]. However, these goals are fixed and do not depend on a state/condition as in self-adaptive systems, described in the *SOTA* approach. The sentence and token level architectures target accurate predictions at each respective level. The two components of the two-phase architecture have local goals of learning each task well, capturing the knowledge (i.e. statistical regularities) at the given granularity (token or sentence), which leads to a global goal of making accurate predictions for the final task using both tasks' knowledge.

To evaluate the achievement of these goals and compare architectures, we use different variants of the F1 score, which are appropriate for sentence and token levels, following CASE 2021 event detection shared task [33]. Generally, F1 is calculated as the weighted harmonic mean of precision and recall. In the below equations, TP, FP and FN refer to the true positive, false positive and false negative counts, respectively.

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{4}$$

For the sentence level evaluations, we use macro averaged F1. It is the unweighted mean of F1 scores calculated per label/class as in Eq. (5). n represents the total number of classes, and $F1_i$ represents the per-class $F1$.

$$Macro\ F1 = \frac{\sum_{i=1}^{n} F1_i}{n} \tag{5}$$

For the token level evaluations, we use the F1 measure introduced with CoNLL 2003 shared task [58]. This score also follows the Eq. (4) but considers text spans and their labels to compute TP, FP and FN values. It marks a span correct only if it exactly matches the actual label.

5.3 Pre-trained transformers

We use three monolingual and one multilingual pre-trained transformer models based on the targeted languages for our experiments. As monolingual models, BERT (*bert-large-cased*) [18], and its variants, BERTimbau (*BERTimbau large*) [59] and BETO (*BETO cased*) [60] models trained in English, Portuguese and Spanish are used. As the

Table 3 Transformer model details, including the number of trained languages (#Lgs.), layers (L), hidden states (H_m), attention heads (A), total parameters (#Params) and the vocabulary size (V). Under tokenisers, SPM refers to Sentence Piece Model, and BPE refers to Byte Pair Encoding

Model	#Lgs.	Tokeniser	L	H_m	A	V	#Params
BERT	1	WordPiece [62]	24	1024	16	30k	335 M
BERTimbau	1	WordPiece [62]	24	1024	16	30k	330 M
BETO	1	BPE [63]	12	1024	16	32k	110 M
XLM-R	100	SPM [63]	24	1024	16	250k	550 M

Table 4 Training data formats with learning strategies

Strategy	Format	Description
Monolingual learning	L_1	Learn from data in language L_1
Language-based TL	$L_1 \rightarrow L_2$	Learn the same task from data in language L_1 and then from data in language L_2
Multilingual learning	$L_1 + L_2 + \cdots + L_n$	Learn from data in multiple languages $L_1, L_2, \ldots L_n$ simultaneously
TTL	$L(1) - L(2)$	Learn the first phase (task 1) from data in language/language combination $L(1)$ and the second phase (task 2) from data in language/language combination $L(2)$

multilingual model, XLM-R (*xlm-roberta-large*) [29] model trained in 100 languages, including the targeted languages is used. Multilingual BERT (mBERT) and XLM-R models were commonly used as multilingual transformers by previous research, but mostly the XLM-R model outperformed the mBERT model, considering its cross-linguality and larger training corpus [5, 29]. Therefore, we only use the XLM-R model for this research. We used HuggingFace's model repository [61] to obtain the pre-trained transformers. Table 3 summarises more details about these models.

5.4 Hyper-parameters

To maintain consistency among architectures to generate comparable results, we used a common set of hyper-parameters for our experiments. For all models, we fixed the maximum sequence length to 128, considering the sequence length distribution of targeted data. Considering the computational complexities associated with transformers, we used the batch size of eight, the learning rate of $1e^{-5}$ with Adam optimiser and epochs of three with early stopping patience of 10. We set evaluation steps allowing 6–13 evaluations per training epoch depending on the size of the training dataset. A split of 10% from training data is used for these evaluations, and the rest is used for training. To mitigate the impact on results by the randomness associated with deep neural networks, we used the majority-class self-ensemble approach [64], following recent trends [5, 19]. With this setting, per experiment, we trained five models initialised with different random seeds and took the majority vote of model predictions as the final prediction.

5.5 Training formats

We involve language-based learning strategies introduced in Sect. 4.3 along with the TTL for our experiments. Thus, we use a common convention to format training data depending on the learning strategy to report our results consistently, as described in Table 4.

6 Results and discussion

This section presents the evaluation results of event sentence identification and trigger and argument extraction using our architectures and the proposed learning strategies. While applying language-based learning strategies, we only allowed transfer and zero-shot learning from high-resource to low-resource languages because the other way is not sensible. Under one-phase learning (Sect. 6.1), we report the results of transformer-based sequence and token classification architectures only learning the corresponding level of data. Also, we address the research questions: RQ1 and RQ2 under this section, analysing both sentence and token level results. Section 6.2 reports the results of two-phase architecture, which learns both sentence and token level data in a sequential manner and answers the RQ3 based on our findings.

6.1 One-phase learning

We report and discuss the results of transformer-based sequence and token classification architectures by learning

Table 5 Sentence level results: macro F1 values using transformer-based sequence classification architecture

Strategy	Transformer	Training data	Language		
			En	Pt	Es
Monolingual learning	BERTimbau	Pt	–	0.7068	–
	BETO	Es	–	–	0.7958
	BERT	En	0.8253	–	–
	XLM-R	Pt	–	NT	–
	XLM-R	Es	–	–	0.4814
	XLM-R	En	0.7900	0.8518‡	0.8121‡
Transfer learning	XLM-R	En→Pt	0.7991	0.8429	0.7547‡
		En→Es	0.8174	**0.8871**‡	0.8199
Multilingual learning	XLM-R	En+Pt	**0.8307**	0.8585	0.7547‡
		En + Es	0.8265	0.8596‡	0.8305
		En + Pt + Es	0.8127	0.8665	**0.8448**

Strategy and *Language* indicate the language-based learning strategy and language of test data. NT shows the models which were not trainable due to data limitations. Zero-shot learning scenarios are marked with ‡, and the best results per language are in bold

one-phase (sentence or token level data) in Sects. 6.1.1 and 6.1.2, respectively.

6.1.1 Event sentence identification

For one-phase learning of event sentence identification, we conducted experiments using transformer-based sequence classification architecture (Fig. 3), involving the language-based learning strategies introduced in Sect. 4.3. To build classifiers, we used monolingual transformers: BERT, BERTimbau and BETO and the multilingual transformer: XLM-R. We refer to the models based on monolingual transformers as monolingual models and the models based on multilingual transformers as multilingual models in the below content for simplicity. The obtained results are reported in Table 5.

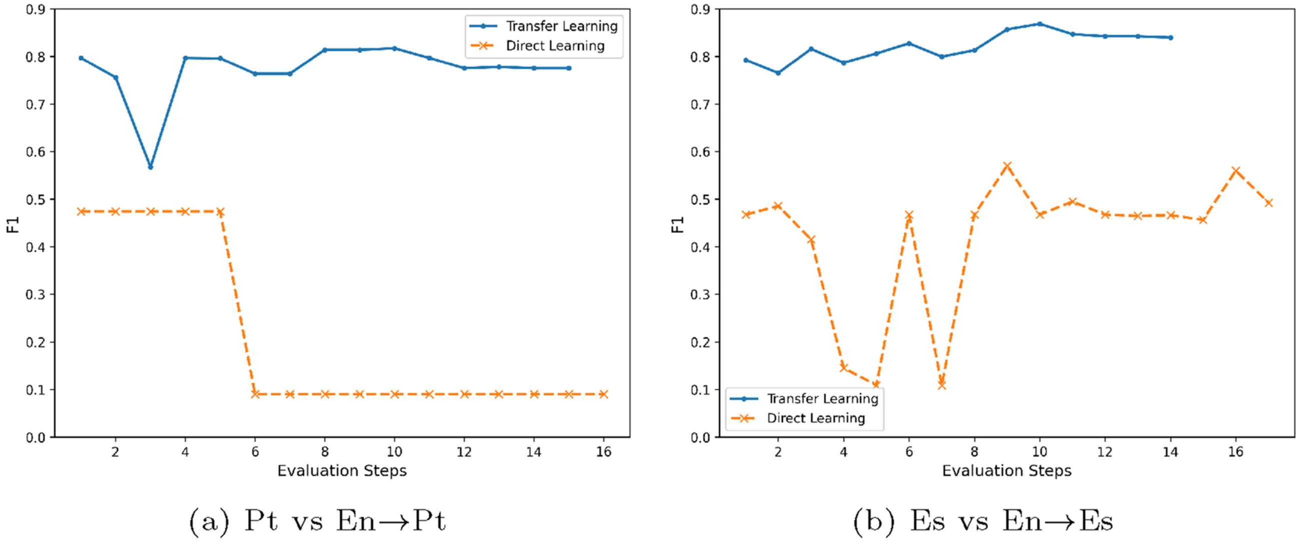
According to results in Table 5, monolingual models trained in a particular language outperformed the multilingual models trained in that language for high-resource (En) and low-resource (Pt and Es) languages. However, with zero-shot learning, the multilingual model trained on the high-resource language made more accurate predictions for low-resource languages than monolingual models. A similar trend is also noticed with the multilingual models, which transfer learned a low-resource language after the high-resource language. The multilingual model performance could be further improved with multilingual learning than with monolingual and multilingual models trained using other learning strategies.

Based on our results, we answer RQ1 and RQ2, focusing on event sentence identification below.

RQ1: *Can an event detection model based on a multilingual transformer, which is only fine-tuned for a particular language, outperform a model based on a monolingual transformer of that language?*

During our experiments, we analysed the performance of models based on monolingual and multilingual transformers for identifying event sentences in three languages (En, Pt and Es). Comparatively, the monolingual transformers we used (i.e. BERT, BERTimbau and BETO) are pre-trained on fewer data than that particular language data used by the multilingual transformer (i.e. XLM-R). However, the larger the vocabulary size, the transformer has a high number of parameters to learn during the fine-tuning (Table 3). Thus, as can be seen in our results in Table 5 under monolingual learning, when a few training instances are available for fine-tuning, monolingual models can learn better in identifying event sentences than the multilingual models, even though the monolingual transformers have seen fewer data during language modelling. For the high-resource language (En) with 22.5k training instances, the monolingual model improved the macro F1 by 3.5% more than the multilingual model. Smaller the training data size, monolingual models showed more improvements than the multilingual model. For Pt, the XLM-R-based model did not converge (behaved as a majority class classifier), but BERTimbau returned 71% macro F1, and for Es, BETO returned 31.4% higher macro F1 than XLM-R.

In summary, multilingual transformer typically requires more training data to fine-tune for the event sentence identification task than the monolingual models, considering the parameter counts. Thus, if data are insufficient for fine-tuning, multilingual models cannot perform better than monolingual models. This claim is further supported by the variations in F1 gaps between multilingual and monolingual models for different languages with different training data sizes mentioned above. Higher the data size, a low gap



(a) Pt vs En→Pt

(b) Es vs En→Es

Fig. 6 Macro F1 scores for the validation sets at different evaluation steps of the sentence level training processes, which involved direct language learning (Pt, Es) and transfer learning from a high-resource language (En → Pt, En → Es) with the sequence classification model with XLM-R transformer

is returned, indicating that the multilingual model can perform on par or better than the monolingual models if enough training data exists, agreeing with the conclusions made by the XLM-R model's original study [29].

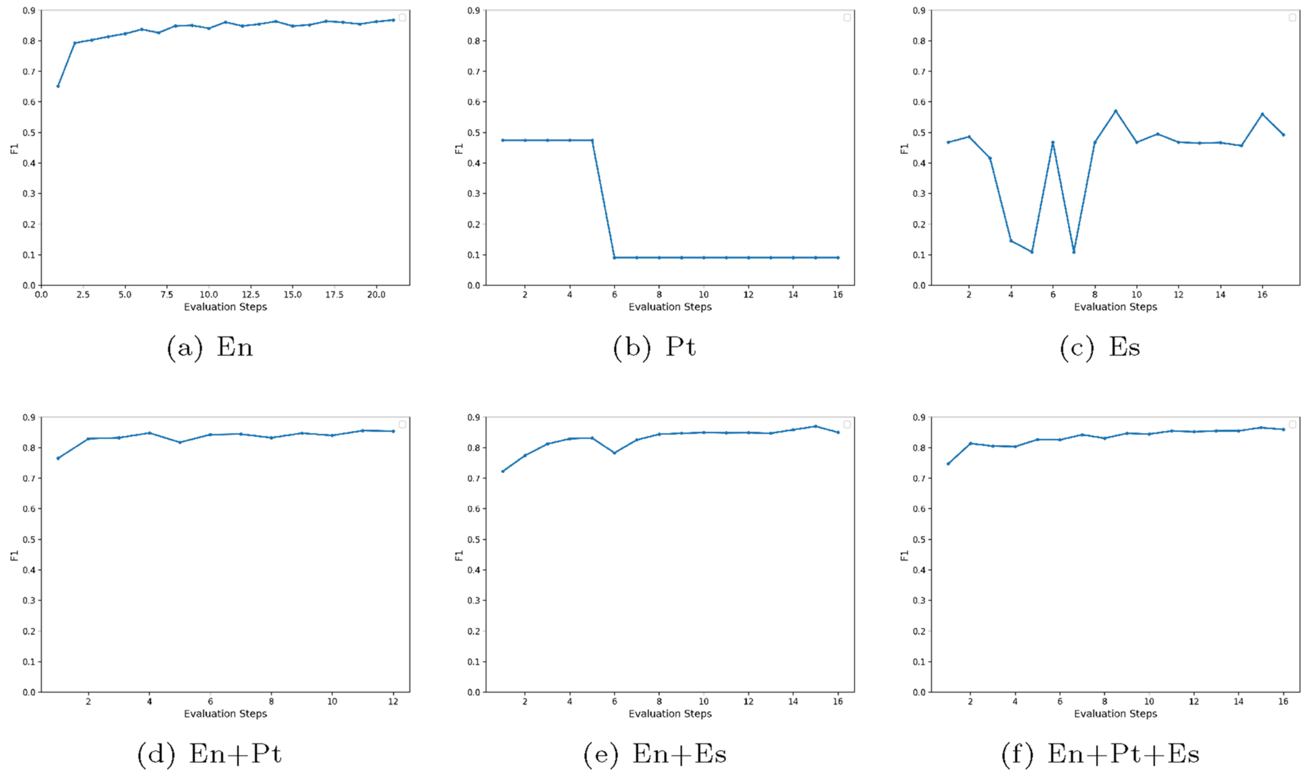
RQ2: *Can a high-resource language improve the event detection performance of a low-resource language using the cross-linguality in transformer models?*

Targeting this question, we involved different learning strategies to analyse the cross-lingual capabilities of the multilingual transformer model we chose (XLM-R). With zero-shot learning, the multilingual sentence classification model, which only learned the high-resource language (En), outperformed the monolingual models that learned corresponding low-resource languages (Table 5). Agreeing with our findings for RQ1, the XLM-R model fine-tunes well when sufficient training instances are provided. Utilising its cross-linguality, effective predictions can make for low-resource languages, learning high-resource languages.

Also, we obtained improved sentence classification results for low-resource languages from multilingual models, which transfer learned the low-resource language (Pt or Es) after the high-resource language (En) (Table 5). As can be seen in Fig. 6, with transfer learning (TL), multilingual models return high macro F1 values from the beginning of evaluation steps, unlike with direct learning. This indicates that even with few training instances, a model can learn well following the knowledge obtained during high-resource language training and the cross-lingual abilities of the transformer. Even for the scenario with Pt where the model did not converge with direct learning due to data limitations, TL returned macro F1 scores around 80% throughout the evaluations emphasising its effectiveness (Fig. 6a). However, no notable improvements are recognised, mostly comparing the multilingual models which transfer learned from the high-resource language and models which only learned the high-resource language. This indicates that if the low-resource language datasets are very small compared to the high-resource language data, they cannot significantly impact the model performance via TL.

Furthermore, we experimented with multilingual learning. Mostly, models fine-tuned using multilingual learning outperformed the models which only learned the high-resource language or transfer learned a low-resource language for sentence level predictions (Table 5). Additionally, Fig. 7 illustrates how macro F1 values vary over evaluation steps with monolingual and multilingual learning. With monolingual learnings, the high-resource language (En) has a high F1 value from the second evaluation step, but other low-resource languages have very low F1 values over all steps. However, with multilingual learning, for all combinations, models return high F1 values (approximately ≥80%) throughout all evaluations (Fig. 7d–f). These results reveal that a cross-lingual model can train well on each language (or adjust its parameters appropriate for multiple languages) when it sees all language data together rather than seeing the languages separately. Also, this way allows the effective utilisation of low-resource language data irrespective of the data size, unlike the scenario with TL.

In summary, these findings lead to a positive answer to RQ2. High-resource languages can improve the event sentence identification performance of low-resource languages using cross-linguality in transformer models.



(a) En

(b) Pt

(c) Es

(d) En+Pt

(e) En+Es

(f) En+Pt+Es

Fig. 7 Macro F1 scores for the validation sets at different evaluation steps of the sentence level training processes, which involved monolingual (En, Pt, Es) and multilingual (En + Pt, En + Es, En + Pt + Es) learning with the sequence classification model with XLM-R transformer. During multilingual learnings, a composition of samples from each language is used as the validation set

Table 6 Token level results: CoNLL 2003 F1 values using transformer-based token classification architecture

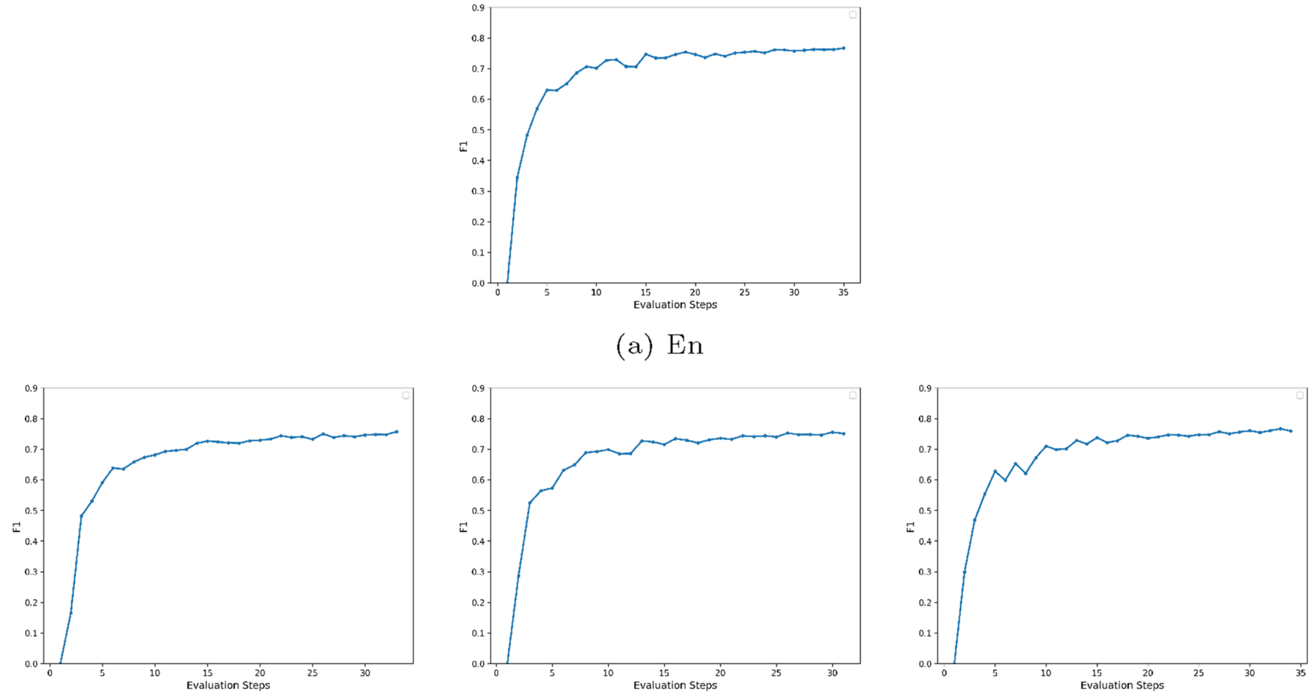
Strategy	Transformer	Training data	Language		
			En	Pt	Es
Monolingual learning	BERT	En	0.7517	–	–
	XLM-R	En	0.7511	0.7043‡	0.6461‡
Multilingual learning	XLM-R	En+Pt	**0.7678**	0.7389	0.6587‡
		En + Es	0.7540	0.7151‡	0.6700
		En + Pt + Es	0.7616	**0.7441**	**0.6752**

Strategy and *Language* indicate the language-based learning strategy and language of test data. Zero-shot learning scenarios are marked with ‡, and the best results per language are in bold

Zero-shot learning can be effectively applied using a multilingual model that is fine-tuned only on high-resource language data for a scenario with no training data available for a low-resource language. When few training instances are available for low-resource languages, a multilingual model can be fine-tuned effectively by combining all the data using multilingual learning, outperforming the language-based TL approach.

6.1.2 Event trigger and argument extraction

For event trigger and argument extraction, we utilised transformer-based token classification architecture (Fig. 4) along with the language-based learning strategies (Sect. 4.3). However, we had to skip a few strategies due to training data limitations. For low-resource languages (Pt and Es), token level data are minimal (<100 instances), and thus, monolingual



(a) En

(b) En+Pt

(c) En+Es

(d) En+Pt+Es

Fig. 8 CoNLL 2003 F1 scores for the validation sets at different evaluation steps of the token level training processes, which involved monolingual (En) and multilingual (En + Pt, En + Es, En + Pt + Es) learning with the token classification model with XLM-R transformer. During multilingual learnings, a composition of samples from each language is used as the validation set

and language-based TL experiments could not be conducted. Therefore, we only require the English transformer model: BERT and the multilingual model: XLM-R for token level experiments. Similar to the above section, we refer to the models based on BERT as monolingual models and models based on XLM-R as multilingual models to maintain consistency and generality of the content. The obtained results are available in Table 6.

Comparatively, token level predictions are less accurate than sentence level predictions, emphasising the complexity of the token level task. According to Table 6 results, for the high-resource language (En), the monolingual model performed slightly better than the multilingual model supporting the claim we made with sentence level results. For low-resource languages, good F1 scores (≥65%) could be obtained with zero-shot learning on the multilingual model trained on the high-resource language. The involvement of multilingual learning further improved the results of high- and low-resource languages, effectively utilising the few labelled instances available with low-resource languages.

Following our results, we answer RQ2, focusing on event trigger and argument extraction below. Due to training data limitations, we could not train monolingual models for low-resource languages to compare with multilingual models and thus skip addressing RQ1 for the token level. However, for En, the monolingual model slightly improved over the multilingual model, which is only fine-tuned using that language, agreeing with our finding for RQ1 based on sentence level results.

RQ2: *Can a high-resource language improve the event detection performance of a low-resource language using the cross-linguality in transformer models?*

Like sentence level analysis, we used different learning strategies with the selected multilingual transformer model (XLM-R) to address this question, focusing on event trigger and argument extraction. However, language-based TL could not be applied since there are no enough training instances from low-resource languages to learn separately. With zero-shot learning, the multilingual token classification model, which only learned the high-resource language (En), returned good results (F1 scores ≥65%) for low-resource languages, as can be seen in our results in Table 6. These results clearly highlight the cross-linguality of the XLM-R model, which can effectively utilise for low-resource language token level predictions with no training data.

The token level results were further improved with multilingual learning (Table 6), similar to our findings with event sentence identification. Multilingual learning allowed the model to learn using the high resource language (En) data and the few training instances of low

Table 7 Sentence level results: macro F1 values using two-phase classification architecture, which learns the sentence level task following the token level task

Strategy	Transfo-mer	Training Data	Language		
			En	Pt	Es
monolingual learning	BERT	En – En	0.8049	-	-
	XLM-R	En – Pt	-	0.4997	-
	XLM-R	En – Es	-	-	0.7638
	XLM-R	En – En	0.7879	0.8543‡	0.8034‡
transfer learning	XLM-R	En – En→Pt	0.6028	0.5568	0.5763‡
		En – En→Es	0.6985	0.8404‡	0.7964
multilingual learning	XLM-R	En – En+Pt	0.8219	0.8646	0.8088‡
		En+Pt – En+Pt	0.8085	0.8658	0.8094‡
		En – En+Es	0.7884	0.8749‡	0.8255
		En+Es – En+Es	**0.8352**	0.8631‡	0.8328
		En – En+Pt+Es	0.7943	0.8708	0.8186
		En+Pt+Es – En+Pt+Es	0.8149	**0.8772**	**0.8404**

Strategy and *Language* indicate the language-based learning strategy and language of test data. Zero-shot learning scenarios are marked with ‡, and the best results per language are in bold. Highlighted cells indicate the improved F1 scores than only learning sentence data

Table 8 Token level results: CoNLL 2003 F1 values using two-phase classification architecture, which learns the token level task following the sentence level task

Strategy	Transfo-mer	Training Data	Language		
			En	Pt	Es
monolingual learning	BERT	En – En	0.7513	-	-
	XLM-R	En – En	0.7566	0.6930‡	0.6266‡
multilingual learning	XLM-R	En – En+Pt	0.7548	0.7222	0.6478‡
		En+Pt – En+Pt	0.7542	0.7275	0.6377‡
		En – En+Es	0.7568	0.7164‡	0.6652
		En+Es – En+Es	0.7525	0.7012‡	0.6567
		En – En+Pt+Es	0.7575	0.7364	0.6620
		En+Pt+Es – En+Pt+Es	**0.7599**	**0.7441**	**0.6780**

Strategy and *Language* indicate the language-based learning strategy and language of test data. Zero-shot learning scenarios are marked with ‡, and the best results per language are in bold. Highlighted cells indicate the improved F1 scores than only learning token data

resource languages (Pt and Es), which are insufficient to build separate models or apply language-based TL. We also analysed how the CoNLL F1 scores vary over the evaluation steps of each learning setting (Fig. 8). However, we do not have monolingual models from each language to compare with. Also, we cannot see clear distinctions in the F1 scores between the En and multilingual models, similar to the sentence level analysis. When low-resource language data are limited, the validation split at each setting is almost identical to the En validation split. Thus, we see nearly constant behaviour of F1 scores across all settings. Even though the improvements are not clearly visible over the evaluation steps of the training phase, the final predictions on test data emphasise the effectiveness of multilingual learning.

In summary, we can also provide a positive answer to RQ2 based on token level results. High-resource languages can improve the event trigger and argument extraction performance of low-resource languages, using the cross-lingual capabilities of transformers. Zero-shot learning can be effectively used in scenarios with no training data. It is effective to use multilingual learning when few training instances are available from low-resource languages, irrespective of their count.

6.2 Two-phase learning

In this section, we report and discuss the results of the TTL approach along with the pre-trained transformer models and language-based learning strategies we involved with one-phase learning. For event sentence identification, we trained the model for the token level task before the sentence level task using the proposed architecture in Fig. 5b. The opposite learning sequence is followed for the event trigger and argument extraction (Fig. 5a). The obtained results are available in Tables 7 and 8.

As can be seen in Table 7, TTL (learning token level task before sentence level task) improved the performance of low-resource language predictions at the sentence level in the majority of cases. Multilingual models trained on the high-resource language token data before training on low-resource language sentence data outperformed the multilingual models, which only learned low-resource language sentence data. Also, the multilingual models, which learned the high-resource language token and sentence data, returned higher F1 values for low-resource languages than the scores of monolingual models, which only learned the sentence level of that particular language. However, combining language-based TL with TTL did not improve the results for any language. Contrarily, with multilingual learning, TTL performed better in most cases than only learning sentence data.

Following the results in Table 8, overall, TTL (learning sentence level task before token level task) did not improve the token level predictions even though more instances are available with sentence data. However, with monolingual learning, the multilingual model performance could improve for the high-resource language with TTL rather than only learning token data. Also, on a few occasions, applying TTL with multilingual learning improved the results compared to the models that only learned token data. Based on the results, we answer RQ3 below.

RQ3: *Can two-phase transfer learning (TTL) on transformers using different event detection tasks improve the performance of involved tasks in monolingual and multilingual settings?*

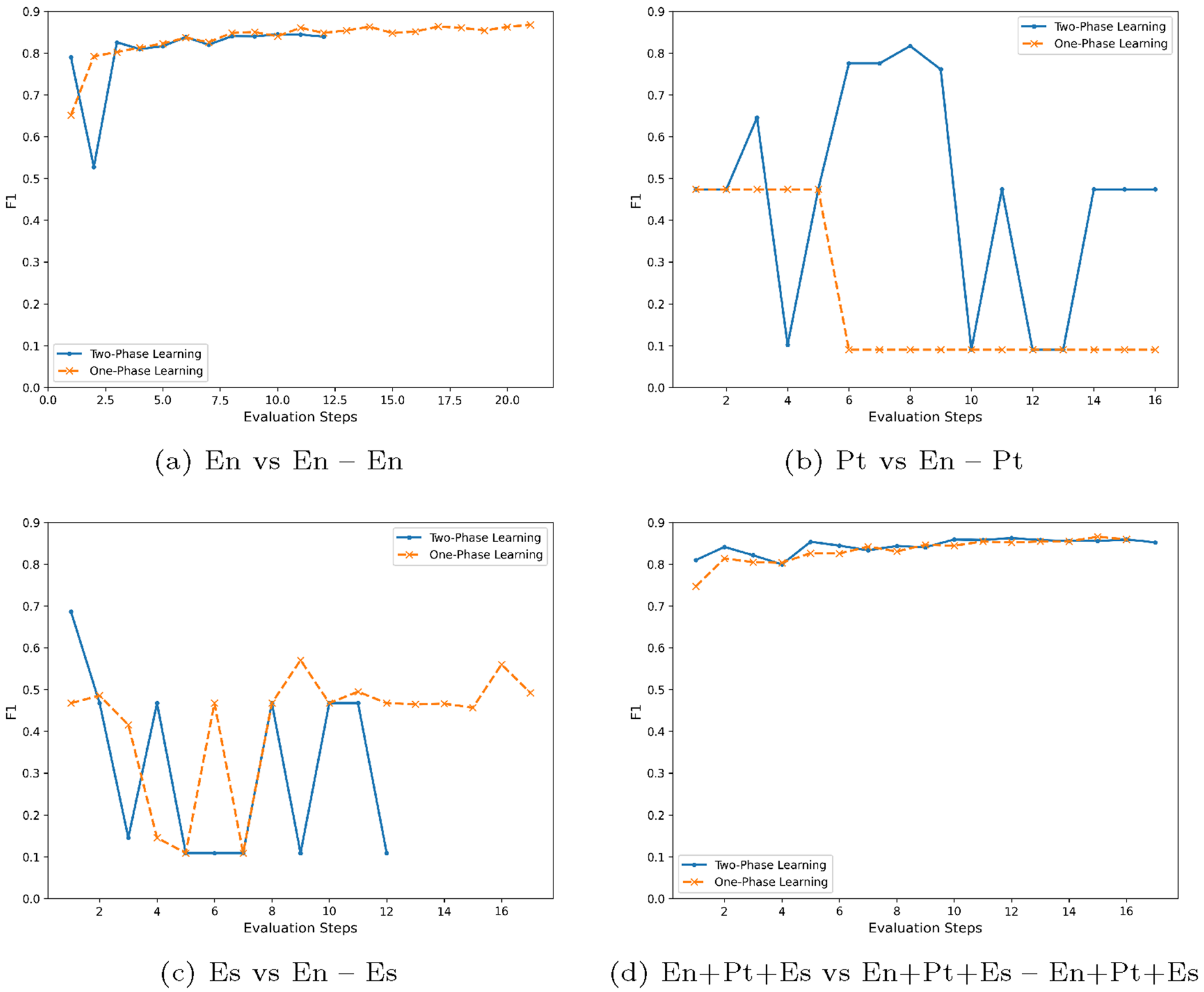
We analysed the performance of TTL involving the tasks: event sentence identification and event trigger and argument extraction at two data granularities: sentence and token level. Our experiments showed improvements in the sentence level predictions in most cases using the models which learned token level data beforehand (Table 7). Further analysis on variations in macro F1 values over model evaluation steps also confirmed that TTL from token level helps sentence level learning. As can be seen in Fig. 9, for monolingual and multilingual learning, the sentence level learning process begins with high F1 scores or achieves high F1 scores in a few steps with TTL than the scores obtained by learning the sentence level task directly. However, in most cases, token level predictions were not improved by learning sentence data beforehand, even though more training instances are available at the sentence level (Table 8). As shown in Fig. 10, during the model training process also, TTL behaves similar to learning token data directly. Since token level labels directly help resolve sentence level labels, learning the token data help the model to improve sentence level predictions. Contrarily, token labels cannot be predicted by seeing sentence labels. Thus, learning sentence labels beforehand does not help the model much with token level predictions, even though the instance count is high.

In summary, TTL can improve the performance of a task in monolingual and multilingual settings by learning a related task that can help derive the targeted labels during the first phase. In other terms, this strategy can mainly be used to improve the performance of a coarse-grained task based on a related fine-grained task. The task-relatedness is more crucial in this learning than the training dataset sizes. This strategy is more helpful in scenarios that require making predictions for low-resource languages with few or no training instances, as data from other languages prepared for related tasks can be used.

However, learning two phases requires more training time or resources than learning one phase. Yet, the training process has no impact on the final model's size and inference time, which are critical for its later usage, as these factors only depend on the model architecture. Our analyses further confirmed this fact, along with the memory usages and inference times reported in Table 9, which are common to a particular transformer model without relying on the targeted task (i.e. sentence or token level prediction) or the fine-tuning process (i.e. one- and two-phase learning). Overall, all built models take less time than a second on a GPU and a maximum of 7 s on a CPU to make a prediction. Therefore, if the training process helps improve the final predictions, the additional time it takes can be neglected, considering the model's later usage for many effective predictions. Additionally, this fast inferencing ability, which can further improve by increasing the machine's computational power, indicates the models' scalability for making predictions on a large data volume within a shorter period.

7 Conclusions and future work

In this paper, we proposed a novel learning strategy named *Two-phase Transfer Learning (TTL)*, allowing transformer models to learn from different levels of data granularity (i.e. sentence and token). Our approach is expandable to any related sentence and token-level task irrespective of its domain or language, as no domain- or language-specific features are involved. Transformers are especially involved in our approach, considering their transferability, cross-linguality, context awareness and state-of-the-art performance in many NLP applications. We applied TTL to news event detection and analysed how it can improve sentence and token level tasks by transferring knowledge in this paper.



(a) En vs En – En

(b) Pt vs En – Pt

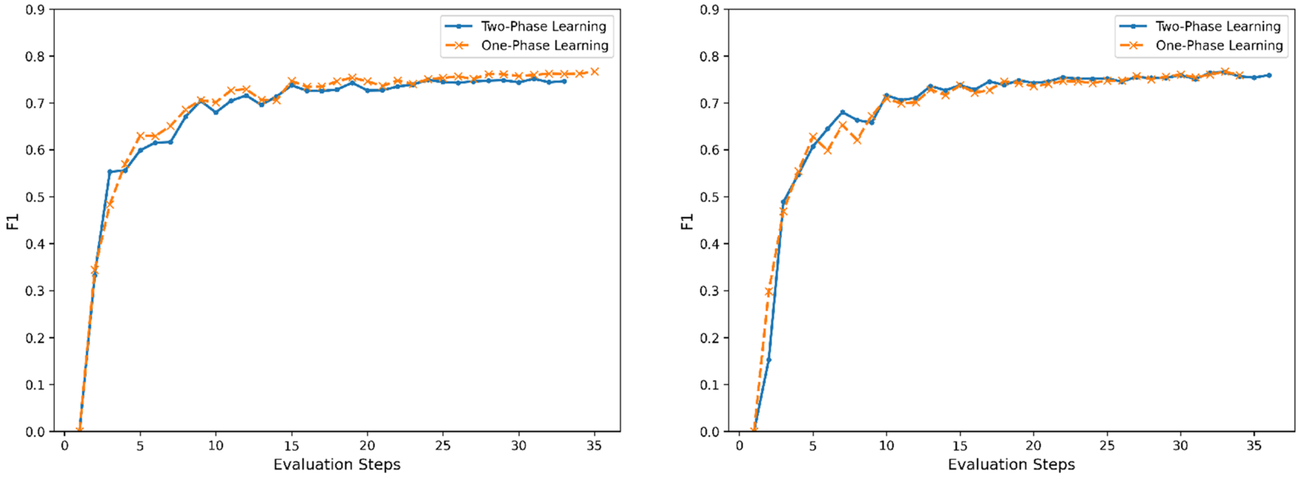
(c) Es vs En – Es

(d) En+Pt+Es vs En+Pt+Es – En+Pt+Es

Fig. 9 Macro F1 scores for the validation sets at different evaluation steps of the sentence level training processes using the sequence classification model (one-phase learning) and two-phase classification model (two-phase learning) with XLM-R transformer. For multilingual learning, a composition of samples from each language is used as the validation set

Also, to the best of our knowledge, this is the first effort to report a comprehensive experimental study on cross-lingual event detection, covering sentence and token level tasks and their transferability.

We used the multilingual version of the GLOCON gold standard dataset and several monolingual and multilingual pre-trained transformer models for our experiments. Our findings show that if sufficient training data exist, a multilingual transformer-based model can outperform a monolingual model, answering RQ1 of this research. Also, our experiments indicate that high-resource languages can improve the event detection performance of low-resource languages, using cross-linguality in transformer models, especially with multilingual and zero-shot learning, addressing RQ2. These findings will be beneficial from the perspective of applications because a multilingual event detection model can cover multiple languages effectively in a resource-efficient manner than having several monolingual models per language. Following RQ3, with the involvement of TTL, we could further improve the model performances in monolingual and multilingual settings. However, the relatedness of tasks is more crucial in this learning than the training data sizes. If the first task can help the second task's predictions, the model can gain some knowledge from the first task to improve the second task's performance through TTL. Thus, we noticed more improvements by learning the sentence level task after the token level task since the token data can help derive the labels of sentences. Additionally, the ability to learn from different language data at different granularities helps



(a) En vs En – En

(b) En+Pt+Es vs En+Pt+Es – En+Pt+Es

Fig. 10 CoNLL 2003 F1 scores for the validation set at different evaluation steps of the token level training process using the token classification model (one-phase learning) and two-phase classification model (two-phase learning) with XLM-R transformer. For multilingual learning, a composition of samples from each language is used as the validation set

Table 9 Memory usage and inference speed of transformer-based models built for news media event detection on GeForce RTX 3090 GPU and Intel(R) Xeon(R) CPU @ 2.30 GHz

Transformer	Disk Usage (MB)	GPU		CPU	
		RAM (MB)	Time (s)	RAM (MB)	Time (s)
BERT	1274	6539	0.0562	3295	3.7190
BERTimbau	1277	6540	0.0560	3312	6.5957
BETO	420	5482	0.0481	1587	1.8834
XLM-R	2150	7680	0.7040	5018	6.9053

build effective models for low-resource languages, utilising available data.

In future work, we plan to extend our research to more languages and analyse how the interconnections between languages can be utilised to improve the performance of event detection tasks. Also, in this work, we only focused on the languages which are supported by available pre-trained transformer models such as XLM-R. To fill this gap, we aim to construct datasets for not supported languages and evaluate their performance in future. Considering TTL, we designed it in a general manner, which is applicable to any related sentence and token level classification tasks, such as sentence and token level predictions in sentiment analysis or offensive language identification, rather than limiting it to event detection. Thus, we also plan to thoroughly investigate TTL's applicability to different domains and research areas.

Funding No funding was received for conducting this study.

Data availability The datasets involved in this study were published by [33] and can be accessed following the instructions on https://github.com/emerging-welfare/case-2021-shared-task

Declarations

Conflict of interest The authors have no competing interests to declare that are relevant to the content of this article.

Code availability The codebase is publicly available on https://github.com/HHansi/MultiEventMiner

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

References

1. Hettiarachchi H, Adedoyin-Olowe M, Bhogal J, Gaber MM (2022) Embed2Detect: temporally clustered embedded words for event detection in social media. Mach Learn 111:49–87. https://doi.org/10.1007/s10994-021-05988-7
2. Balali A, Asadpour M, Campos R, Jatowt A (2020) Joint event extraction along shortest dependency paths using graph convolutional networks. Knowl-Based Syst 210:106492. https://doi.org/10.1016/j.knosys.2020.106492

3. Sha L, Qian F, Chang B, Sui Z (2018) Jointly extracting event triggers and arguments by dependency-bridge RNN and tensor-based argument interaction. In: Proceedings of the AAAI conference on artificial intelligence, vol 32(1)
4. Hürriyetoğlu A, Yörük E, Mutlu O, Duruşan F, Yoltar Ç, Yüret D, Gürel B (2021) Cross-context news corpus for protest event-related knowledge base construction. Data Intell 3(2):308–335. https://doi.org/10.1162/dint_a_00092
5. Hettiarachchi H, Adedoyin-Olowe M, Bhogal J, Gaber MM (2021) DAAI at CASE 2021 task 1: Transformer-based multilingual socio-political and crisis event detection. In: Proceedings of the 4th workshop on challenges and applications of automated extraction of socio-political events from text (CASE 2021), pp 120–130. Association for Computational Linguistics. https://doi.org/10.18653/v1/2021.case-1.16. https://aclanthology.org/2021.case-1.16
6. Naughton M, Stokes N, Carthy J (2010) Sentence-level event classification in unstructured texts. Inf Retr 13(2):132–156. https://doi.org/10.1007/s10791-009-9113-0
7. Hong Y, Zhang J, Ma B, Yao J, Zhou G, Zhu Q (2011) Using cross-entity inference to improve event extraction. In: Proceedings of the 49th annual meeting of the association for computational linguistics: human language technologies. Association for Computational Linguistics, Portland, Oregon, USA, pp 1127–1136. https://aclanthology.org/P11-1113
8. Chen C, Ng V (2012) Joint modeling for Chinese event extraction with rich linguistic features. In: Proceedings of COLING 2012. The COLING 2012 Organizing Committee, Mumbai, India, pp 529–544. https://aclanthology.org/C12-1033
9. Chen Y, Xu L, Liu K, Zeng D, Zhao J (2015) Event extraction via dynamic multi-pooling convolutional neural networks. In: Proceedings of the 53rd annual meeting of the association for computational linguistics and the 7th international joint conference on natural language processing (vol 1: long papers), pp 167–176. Association for Computational Linguistics, Beijing, China. https://doi.org/10.3115/v1/P15-1017. https://aclanthology.org/P15-1017
10. Hassan A, Mahmood A (2017) Deep learning for sentence classification. In: 2017 IEEE long island systems, applications and technology conference (LISAT), pp 1–5. https://doi.org/10.1109/LISAT.2017.8001979
11. Pandey C, Ibrahim Z, Wu H, Iqbal E, Dobson R (2017) Improving RNN with Attention and embedding for adverse drug reactions. In: Proceedings of the 2017 international conference on digital health. DH ’17. Association for Computing Machinery, New York, NY, USA, pp 67–71. https://doi.org/10.1145/3079452.3079501
12. Liu S, Li Y, Zhang F, Yang T, Zhou X (2019) Event detection without triggers. In: Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, vol 1 (long and short papers). Association for Computational Linguistics, Minneapolis, Minnesota, pp 735–744. https://doi.org/10.18653/v1/N19-1080. https://aclanthology.org/N19-1080
13. Alyafeai Z, AlShaibani MS, Ahmad I (2020) A survey on transfer learning in natural language processing. arXiv preprint arXiv:2007.04239
14. Dumoulin V, Houlsby N, Evci U, Zhai X, Goroshin R, Gelly S, Larochelle H (2021) Comparing transfer and meta learning approaches on a unified few-shot classification benchmark. arXiv preprint arXiv:2104.02638
15. Chowdhury A, Chaudhari D, Chaudhuri S, Jermaine C (2022) Meta-meta classification for one-shot learning. In: 2022 IEEE/CVF winter conference on applications of computer vision (WACV), pp 1628–1637. https://doi.org/10.1109/WACV51458.2022.00169
16. Ruder S, Peters M.E, Swayamdipta S, Wolf T (2019) Transfer learning in natural language processing. In: Proceedings of the 2019 conference of the north american chapter of the association for computational linguistics: tutorials. Association for Computational Linguistics, Minneapolis, Minnesota, pp 15–18. https://doi.org/10.18653/v1/N19-5004. https://aclanthology.org/N19-5004
17. Chowdhury A, Jiang M, Chaudhuri S, Jermaine C (2021) Few-shot image classification: just use a library of pre-trained feature extractors and a simple classifier. In: 2021 IEEE/CVF international conference on computer vision (ICCV), pp 9425–9434. https://doi.org/10.1109/ICCV48922.2021.00931
18. Devlin J, Chang M-W, Lee K, Toutanova K (2019) BERT: pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, vol 1 (long and short papers). Association for Computational Linguistics, Minneapolis, Minnesota, pp 4171–4186. https://doi.org/10.18653/v1/N19-1423. https://aclanthology.org/N19-1423
19. Awasthy P, Ni J, Barker K, Florian R (2021) IBM MNLP IE at CASE 2021 task 1: multigranular and multilingual event detection on protest news. In: Proceedings of the 4th workshop on challenges and applications of automated extraction of socio-political events from text (CASE 2021). Association for Computational Linguistics, pp 138–146. https://doi.org/10.18653/v1/2021.case-1.18. https://aclanthology.org/2021.case-1.18
20. Lefever E, Hoste V (2016) A classification-based approach to economic event detection in Dutch news text. In: Proceedings of the tenth international conference on language resources and evaluation (LREC’16). European Language Resources Association (ELRA), Portorož, Slovenia, pp 330–335. https://aclanthology.org/L16-1051
21. Basile A, Caselli T (2020) Protest event detection: when task-specific models outperform an event-driven method. In: Lecture notes in computer science. Springer, pp 97–111. https://doi.org/10.1007/978-3-030-58219-7_9
22. Hassan A, Mahmood A (2018) Convolutional recurrent deep learning model for sentence classification. IEEE Access 6:13949–13957. https://doi.org/10.1109/ACCESS.2018.2814818
23. Hochreiter S, Schmidhuber J (1997) Long short-term memory. Neural Comput 9(8):1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735
24. Lawrence S, Giles CL, Tsoi AC, Back AD (1997) Face recognition: a convolutional neural-network approach. IEEE Trans Neural Netw 8(1):98–113. https://doi.org/10.1109/72.554195
25. Huynh T, He Y, Willis A, Rueger S (2016) Adverse drug reaction classification with deep neural networks. In: Proceedings of COLING 2016, the 26th international conference on computational linguistics: technical papers. The COLING 2016 Organizing Committee, Osaka, Japan, pp 877–887. https://aclanthology.org/C16-1084
26. Gürel A, Emin E (2021) ALEM at CASE 2021 task 1: multilingual text classification on news articles. In: Proceedings of the 4th workshop on challenges and applications of automated extraction of socio-political events from text (CASE 2021). Association for Computational Linguistics, pp 147–151. https://doi.org/10.18653/v1/2021.case-1.19. https://aclanthology.org/2021.case-1.19
27. Hu T, Team SN (2021) “No Conflict” at CASE 2021 task 1: pre-training for sentence-level protest event detection. In: Proceedings of the 4th workshop on challenges and applications of automated extraction of socio-political events from text (CASE 2021). Association for Computational Linguistics, pp 152–160. https://doi.org/10.18653/v1/2021.case-1.20. https://aclanthology.org/2021.case-1.20

28. Liu Y, Ott M, Goyal N, Du J, Joshi M, Chen D, Levy O, Lewis M, Zettlemoyer L, Stoyanov V (2019) Roberta: a robustly optimized bert pretraining approach. arXiv preprint arXiv:1907.11692
29. Conneau A, Khandelwal K, Goyal N, Chaudhary V, Wenzek G, Guzmán F, Grave E, Ott M, Zettlemoyer L, Stoyanov V (2020) Unsupervised cross-lingual representation learning at scale. In: Proceedings of the 58th annual meeting of the association for computational linguistics, pp 8440–8451. Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.acl-main.747. https://aclanthology.org/2020.acl-main.747
30. Re F, Vegh D, Atzenhofer D, Team SN (2021) "DaDeFrNi" at CASE 2021 task 1: document and sentence classification for protest event detection. In: Proceedings of the 4th workshop on challenges and applications of automated extraction of socio-political events from text (CASE 2021). Association for Computational Linguistics, pp 171–178. https://doi.org/10.18653/v1/2021.case-1.22. https://aclanthology.org/2021.case-1.22
31. Kalyan P, Reddy D, Hande A, Priyadharshini R, Sakuntharaj R, Chakravarthi BR (2021) IIITT at CASE 2021 task 1: leveraging pretrained language models for multilingual protest detection. In: Proceedings of the 4th workshop on challenges and applications of automated extraction of socio-political events from text (CASE 2021). Association for Computational Linguistics, pp 98–104. https://doi.org/10.18653/v1/2021.case-1.13. https://aclanthology.org/2021.case-1.13
32. Çelik F, Dalkılıç T, Beyhan F, Yeniterzi R (2021) SU-NLP at CASE 2021 task 1: protest news detection for English. In: Proceedings of the 4th workshop on challenges and applications of automated extraction of socio-political events from text (CASE 2021). Association for Computational Linguistics, pp 131–137. https://doi.org/10.18653/v1/2021.case-1.17. https://aclanthology.org/2021.case-1.17
33. Hürriyetoğlu A, Mutlu O, Yörük E, Liza FF, Kumar R, Ratan S (2021) Multilingual protest news detection—shared task 1, CASE 2021. In: Proceedings of the 4th workshop on challenges and applications of automated extraction of socio-political events from text (CASE 2021), pp 79–91. Association for Computational Linguistics. https://doi.org/10.18653/v1/2021.case-1.11. https://aclanthology.org/2021.case-1.11
34. Li Q, Ji H, Huang L (2013) Joint event extraction via structured prediction with global features. In: Proceedings of the 51st annual meeting of the association for computational linguistics (volume 1: long papers). Association for Computational Linguistics, Sofia, Bulgaria, pp 73–82. https://aclanthology.org/P13-1008
35. M'hamdi M, Freedman M, May J (2019) Contextualized cross-lingual event trigger extraction with minimal resources. In: Proceedings of the 23rd conference on computational natural language learning (CoNLL). Association for Computational Linguistics, Hong Kong, China, pp 656–665. https://doi.org/10.18653/v1/K19-1061. https://aclanthology.org/K19-1061
36. Lu S, Li S, Xu Y, Wang K, Lan H, Guo J (2022) Event detection from text using path-aware graph convolutional network. Appl Intell 52(5):4987–4998. https://doi.org/10.1007/s10489-021-02695-7
37. Nguyen TH, Cho K, Grishman R (2016) Joint event extraction via recurrent neural networks. In: Proceedings of the 2016 conference of the North American chapter of the Association for Computational Linguistics: human language technologies. Association for Computational Linguistics, San Diego, California, pp 300–309. https://doi.org/10.18653/v1/N16-1034. https://aclanthology.org/N16-1034
38. Yang S, Feng D, Qiao L, Kan Z, Li D (2019) Exploring pretrained language models for event extraction and generation. In: Proceedings of the 57th annual meeting of the association for computational linguistics. Association for Computational Linguistics, Florence, Italy, pp 5284–5294. https://doi.org/10.18653/v1/P19-1522. https://aclanthology.org/P19-1522
39. Vivek Kalyan S, Paul T, Shaun T, Andrews M (2021) Handshakes AI research at CASE 2021 task 1: exploring different approaches for multilingual tasks. In: Proceedings of the 4th workshop on challenges and applications of automated extraction of socio-political events from text (CASE 2021). Association for Computational Linguistics, pp 92–97. https://doi.org/10.18653/v1/2021.case-1.12. https://aclanthology.org/2021.case-1.12
40. Nugent T, Petroni F, Raman N, Carstens L, Leidner JL (2017) A comparison of classification models for natural disaster and critical event detection from news. In: 2017 IEEE international conference on big data (big data), pp 3750–3759. https://doi.org/10.1109/BigData.2017.8258374
41. Allan J, Papka R, Lavrenko V (1998) On-line new event detection and tracking. In: Proceedings of the 21st annual international ACM SIGIR conference on research and development in information retrieval. SIGIR '98. Association for Computing Machinery, New York, NY, USA, pp 37–45. https://doi.org/10.1145/290941.290954
42. Lin Y, Ji H, Huang F, Wu L (2020) A joint neural model for information extraction with global features. In: Proceedings of the 58th annual meeting of the Association for Computational Linguistics. Association for Computational Linguistics, pp 7999–8009. https://doi.org/10.18653/v1/2020.acl-main.713. https://aclanthology.org/2020.acl-main.713
43. Ranasinghe T, Orasan C, Mitkov R (2020) TransQuest: translation quality estimation with cross-lingual transformers. In: Proceedings of the 28th international conference on computational linguistics. International Committee on Computational Linguistics, Barcelona, Spain, pp 5070–5081. https://doi.org/10.18653/v1/2020.coling-main.445. https://aclanthology.org/2020.coling-main.445
44. Gao C, Zhang X, Liu H, Yun W (2022) A joint extraction model of entities and relations based on relation decomposition. Int J Mach Learn Cybernet 13:1833–1845. https://doi.org/10.1007/s13042-021-01491-6
45. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser L, Polosukhin I (2017) Attention is all you need. In: Guyon I, Luxburg UV, Bengio S, Wallach H, Fergus R, Vishwanathan S, Garnett R (eds) Advances in neural information processing systems, vol 30. Curran Associates Inc., New York
46. Weiss K, Khoshgoftaar TM, Wang D (2016) A survey of transfer learning. J Big data 3(1):1–40. https://doi.org/10.1186/s40537-016-0043-6
47. He X, Chen Y, Ghamisi P (2020) Heterogeneous transfer learning for hyperspectral image classification based on convolutional neural network. IEEE Trans Geosci Remote Sens 58(5):3246–3263. https://doi.org/10.1109/TGRS.2019.2951445
48. Zhuang F, Qi Z, Duan K, Xi D, Zhu Y, Zhu H, Xiong H, He Q (2021) A comprehensive survey on transfer learning. Proc IEEE 109(1):43–76. https://doi.org/10.1109/JPROC.2020.3004555
49. Day O, Khoshgoftaar TM (2017) A survey on heterogeneous transfer learning. J Big Data 4(1):1–42. https://doi.org/10.1186/s40537-017-0089-0
50. Shi X, Liu Q, Fan W, Yu PS, Zhu R (2010) Transfer learning on heterogenous feature spaces via spectral transformation. In: 2010 IEEE international conference on data mining, pp 1049–1054. https://doi.org/10.1109/ICDM.2010.65
51. Moon S, Carbonell J (2016) Proactive transfer learning for heterogeneous feature and label spaces. In: Joint European conference on machine learning and knowledge discovery in databases. Springer, pp 706–721
52. Cruz JCB, Tan JA, Cheng C (2020) Localization of fake news detection via multitask transfer learning. In: Proceedings of the 12th language resources and evaluation conference. European

Language Resources Association, Marseille, France, pp 2596–2604. https://aclanthology.org/2020.lrec-1.316

53. Mathew B, Saha P, Yimam S.M, Biemann C, Goyal P, Mukherjee A (2021) Hatexplain: a benchmark dataset for explainable hate speech detection. In: Proceedings of the AAAI conference on artificial intelligence, vol 35, pp 14867–14875
54. Zhang Y, Yang Q (2021) A survey on multi-task learning. IEEE Trans Knowl Data Eng 34(12):5586–5609. https://doi.org/10.1109/TKDE.2021.3070203
55. Hettiarachchi H, Ranasinghe T (2021) TransWiC at SemEval-2021 task 2: transformer-based multilingual and cross-lingual word-in-context disambiguation. In: Proceedings of the 15th international workshop on semantic evaluation (SemEval-2021). Association for Computational Linguistics, pp 771–779. https://doi.org/10.18653/v1/2021.semeval-1.102. https://aclanthology.org/2021.semeval-1.102
56. Ranasinghe T, Zampieri M (2020) Multilingual offensive language identification with cross-lingual embeddings. In: Proceedings of the 2020 conference on empirical methods in natural language processing (EMNLP). Association for Computational Linguistics, pp 5838–5844. https://doi.org/10.18653/v1/2020.emnlp-main.470. https://aclanthology.org/2020.emnlp-main.470
57. Abeywickrama DB, Bicocchi N, Mamei M, Zambonelli F (2020) The sota approach to engineering collective adaptive systems. Int J Softw Tools Technol Transf 22(4):399–415. https://doi.org/10.1007/s10009-020-00554-3
58. Tjong Kim Sang EF, De Meulder F (2003) Introduction to the CoNLL-2003 shared task: language-independent named entity recognition. In: Proceedings of the seventh conference on natural language learning at HLT-NAACL 2003, pp 142–147. https://aclanthology.org/W03-0419
59. Souza F, Nogueira R, Lotufo R (2020) BERTimbau: pretrained BERT models for Brazilian Portuguese. In: 9th Brazilian conference on intelligent systems, BRACIS, Rio Grande do Sul, Brazil, October 20–23 (to appear). Springer, Berlin, Heidelberg, pp 403–417. https://doi.org/10.1007/978-3-030-61377-8_28
60. Canete J, Chaperon G, Fuentes R, Ho J.-H, Kang H, Pérez J (2020) Spanish pre-trained bert model and evaluation data. In: PML4DC at ICLR 2020
61. Wolf T, Debut L, Sanh V, Chaumond J, Delangue C, Moi A, Cistac P, Rault T, Louf R, Funtowicz M, Davison J, Shleifer S, von Platen P, Ma C, Jernite Y, Plu J, Xu C, Le Scao T, Gugger S, Drame M, Lhoest Q, Rush A (2020) Transformers: state-of-the-art natural language processing. In: Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations. Association for Computational Linguistics, pp 38–45. https://doi.org/10.18653/v1/2020.emnlp-demos.6. https://aclanthology.org/2020.emnlp-demos.6
62. Wu Y, Schuster M, Chen Z, Le Q.V, Norouzi M, Macherey W, Krikun M, Cao Y, Gao Q, Macherey K, Klingner J, Shah A, Johnson M, Liu X, Kaiser L, Gouws S, Kato Y, Kudo T, Kazawa H, Stevens K, Kurian G, Patil N, Wang W, Young C, Smith J, Riesa J, Rudnick A, Vinyals O, Corrado G, Hughes M, Dean J (2016) Google's neural machine translation system: bridging the gap between human and machine translation. CoRR arXiV:1609.08144
63. Kudo T Richardson J (2018) Sentence piece: a simple and language independent subword tokenizer and detokenizer for neural text processing. In: Proceedings of the 2018 conference on empirical methods in natural language processing: system demonstrations. Association for Computational Linguistics, Brussels, Belgium, pp 66–71. https://doi.org/10.18653/v1/D18-2012. https://aclanthology.org/D18-2012
64. Hettiarachchi H Ranasinghe T (2020) InfoMiner at WNUT-2020 task 2: transformer-based covid-19 informative tweet extraction. In: Proceedings of the sixth workshop on noisy user-generated text (W-NUT 2020). Association for Computational Linguistics, pp 359–365. https://doi.org/10.18653/v1/2020.wnut-1.49. https://aclanthology.org/2020.wnut-1.49

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.