

Token-level Identification of Multiword Expressions using Pre-Trained Multilingual Language Models

by

Raghuraman Swaminathan

Bachelor of Technology in Information Technology, Vellore Institute of Technology, 2020

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

Master of Computer Science

In the Graduate Academic Unit of Computer Science

Supervisor(s): Paul Cook, PhD, Computer Science
Examining Board: Huajie Zhang, PhD, Computer Science, Chair
Francis Palma, PhD, Computer Science
Dhirendra Shukla, PhD, Technology Management and Entrepreneurship

This thesis is accepted by the
Dean of Graduate Studies

THE UNIVERSITY OF NEW BRUNSWICK

September, 2023

© Raghuraman Swaminathan, 2023

Abstract

Multiword expressions (MWEs) are combinations of words where the meaning of the expression cannot be derived from its component words. MWEs are commonly used in different languages and are difficult to identify. For different NLP tasks such as sentiment analysis and machine translation, it is important that language models automatically identify and classify these MWEs. While considerable work has been done in identifying and classifying MWEs, little work has been done in a cross-lingual setting. In this thesis, we consider novel cross-lingual settings for MWE identification and idiomaticity prediction in which systems are tested on languages that are unseen during training. We use multilingual models of BERT, specifically mBERT, RoBERTa and mDeBERTa. Our findings indicate that pre-trained multilingual language models are able to learn knowledge about MWEs and idiomaticity that is not language-specific. Moreover, we find that training data from other languages can be leveraged to give improvements over monolingual models.

Acknowledgements

There are many people who have supported me throughout this journey. I would like to acknowledge a few of them without whom this thesis would have not been possible. First and foremost, and the most important people in my life, my family. I would like to thank my parents for always listening to me and pushing me no matter what the hardships. I am grateful for my brother Siva for always being by my side. Secondly, I would like to acknowledge my supervisor, Dr. Paul Cook. His support and knowledge has been invaluable, throughout this two years. I would also like to thank him for providing me the opportunity to publish my first paper and present it at my first conference. Lastly, I would like to thank my friends Bhanu and Digambar, for always cheering me up during periods of difficulty and making this journey memorable.

Table of Contents

Abstract	ii
Acknowledgments	iii
Table of Contents	iv
List of Tables	vi
List of Figures	vii
1 Introduction	1
2 Related Work	7
2.1 Word embeddings	7
2.1.1 Count-based word embeddings	8
2.1.2 Prediction-based word embeddings	9
2.2 BERT	15
2.2.1 RNN	15
2.2.2 Transformers	17
2.2.3 BERT architecture	19
2.3 Cross-lingual	24
2.4 Multiword Expressions	26
3 Models	31
3.1 SemEval	31

3.2	PARSEME	31
4	Materials and Methods	33
4.1	Datasets	33
4.1.1	SemEval	33
4.1.2	PARSEME	35
4.2	Experimental setup	37
4.3	Implementation and parameter settings	39
4.4	Evaluation metrics	40
5	Results	42
5.1	SemEval	42
5.2	PARSEME	44
6	Conclusions	48
	Bibliography	72
	Vita	

List of Tables

4.1	Examples from the SemEval 2022 task 2 subtask A dataset, which are lightly edited to make them more concise.	34
4.2	The number of training and testing instances in the SemEval dataset.	35
4.3	Example of an input for the PARSEME task dataset in french, where the sentence has been lightly modified to make it more concise.	36
4.4	Number of training, development and testing examples in the PARSEME dataset for each language and an average for all languages.	37
5.1	Macro-average F1 score for each model, training and testing on the indicated language(s). Results for a most-frequent class baseline are also shown.	43
5.2	MWE-based, token-based, and unseen F1 score for the monolingual (mono), “all”, and “heldout”, experimental settings, for each language. The “Average” across languages is also calculated.	45
5.3	Frequency of each category and per-category MWE-based F1 score across languages which have instances of these categories.	46

List of Figures

2.1	Example of a co-occurrence matrix [26].	8
2.2	Architecture of CBOW [58].	11
2.3	Architecture of skip-gram [58].	12
2.4	Illustrated example of word embeddings of words “King”, “Queen”, “Man” and “Woman” projected in a 2D space [98].	14
2.5	Architecture of the RNN-based language model [59].	16
2.6	Working diagram of the attention mechanism [4].	17
2.7	Model of the transformer architecture [106].	19
2.8	BERT input representation [24].	20
2.9	Pre-training and fine-tuning stages of the BERT framework [24]. The architecture is similar in both stages, except for the output layers. The output tokens used in the fine-tuning stage depend on the task it is being fine-tuned for.	22
2.10	Masked language modelling vs translation language modelling [49]. . .	25
2.11	monolingual word embeddings (left) and cross-lingual word embed- dings (right) [81].	26
2.12	Architecture of MTLB-STRUCT [100]. The left side of the image contains the input to the BERT model used. The right side contains two different parts which are the layers on top of BERT. The first is a linear layer used for MWE tagging and the second is Tree CRF used to improve the performance of the model.	30

Chapter 1

Introduction

Natural language processing (NLP) is a branch of artificial intelligence that deals with how computer systems understand human languages. One of the earliest works in NLP was the Turing test proposed by Alan Turing who was a British mathematician. The Turing test was designed to test whether computer systems had evolved to the point where it could trick humans into believing they were conversing with other humans [105]. A major step in NLP was the introduction of ELIZA in 1967, a chatbot designed to converse with humans as a therapist [108]. With the advancement in computer systems and the availability of large textual datasets, the use of statistics in NLP was significant starting in the late 1980's [18]. This allowed the systems to understand patterns and improve its knowledge of human languages. Recently, with the advancements in neural networks and transformer-based models, language models have gained an even deeper understanding of human languages, to the point where they outperform humans in some tasks such as next-token prediction [96].

This knowledge gained by computers can be used for a multitude of tasks such as text classification where given some text, the model has to assign a label or class to that text. An example of this is sentiment analysis where the model detects whether a sentence carries a positive, negative or neutral sentiment. Another task is machine

translation which is the automatic translation from one language to another by a system.

An important part to performing these tasks is to understand the meaning of a sentence based on the meanings of its words. However, sentences often contain expressions where the meaning of the expression cannot be derived from the literal meanings of its components. Such expressions are known as multiword expressions (MWEs). MWEs are combinations of lexical items that exhibit some degree of idiomaticity [6]. E.g., *ivory tower* exhibits semantic idiomaticity because its meaning of a place where people are isolated from real-world problems is not transparent from the literal meanings of its component words. For these expressions, there are two major tasks associated with them which can help NLP systems better understand MWEs: 1. Classification: which classifies MWEs as either literal or idiomatic. 2. Identification: which identifies which words in a sentence form MWEs. This can enhance the performance of language models performing downstream tasks such as machine translation [14] and opinion mining [10]. Much work has therefore focused on recognizing MWEs in context, by identifying which tokens in a text correspond to MWEs [31, 77, 78, 93] and by distinguishing idiomatic and literal usages of potentially-idiomatic expressions [28, 34, 43, 48, 54, 85].

Multiword expressions are divided into different types, such as light verb constructions (LVCs) (eg., *take a hike, had a look*), verb particle constructions (VPCs) (eg., *eat up, took out*) and noun compounds (NCs) (eg., *bass player*). These different types often appear in our daily vocabulary and can cause ambiguity to NLP systems when performing tasks where the training data or testing data contains MWEs if they do not incorporate knowledge of MWEs [84]. NLP systems often face problems when understanding MWEs such as the overgeneralization problem wherein certain MWEs have a common word (eg., *telephone box, telephone booth and telephone closet*, where telephone is the common word). For example, expressions such

as *telephone box* and *telephone booth*, where the meaning can be derived from their components, both mean an item that you can make phone calls with, but *telephone closet* is a storeroom where electrical equipment is stored which deviates from its literal meaning [84]. Another common problem that NLP systems face is the lexical-proliferation problem, which refers to when multiple expressions have a common pattern (eg., *take a flight*, *take a hike*) but some of them are compositional in nature and some of are non-compositional in nature such as LVCs which often appears in families such as *take a hike and take a flight* [84]. For example, to take a hike can have the idiomatic meaning of ‘go away’ while take a flight means to catch a flight. When NLP systems encounter these LVCs, they face difficulties understanding the meaning of these LVCs without prior knowledge of them.

Most expressions are compositional in nature but MWEs are a challenge to work with because they can be non-compositional in nature. Compositionality refers to whether a meaning of a MWE can be derived from its component words. MWEs can either be compositional or non-compositional. For example, we can derive the meaning of *kick the bucket* from its component words *kick*, *the* and *bucket* which means to literally kick the bucket (e.g., with one’s foot). In the above example, *kick the bucket* can also have another meaning which is an idiom meaning “to die”. This exhibits non-compositionality where the meaning of the phrase cannot be derived directly from its component words. Automatically identifying non-compositional MWEs has been shown to be a harder task as compared to compositional MWEs [67]. [67] also showed that word embedding models such as word2vec and fastText were better at identifying non-compositional MWEs compared to newer contextual word embedding models such as BERT on some tasks. Later experiments by [97] have shown that contextual embedding models perform better on other tasks.

While most datasets that are used for MWE identification and classification are predominantly in English, there are a few datasets that are in other languages such

as Portuguese [23] and German [91] and recently multilingual datasets have been created such as PARSEME [88, 77, 78]. For other languages, monolingual language models were created for downstream tasks on those languages such as AraBERT [3] for Arabic and RuBERT [47] for Russian, but these models would not be ideal for cross-lingual experiments where training data is in one language and testing data is another. For cross-lingual tasks, multilingual models such as mBERT [24] would be ideal.

One interesting line of investigation is the ability of models to generalize to expressions that were not observed during training. For example, this was a focus in the evaluation of [78]. [27] further explore the ability of language models to encode information about idiomaticity that is not specific to a particular language by considering cross-lingual idiomaticity prediction, in which the idiomaticity of expressions in a language that was not observed during training is predicted.

In our current study, we have two research questions. The first one is *Can models for automatically predicting idiomaticity generalize to MWEs in a different language that was not seen during training?* The results in chapter 5 indicate that models are able to learn information about idiomaticity that is not language-specific. The *heldout* setting in Table 5.2 show that models such as mBERT [24] are able to identify unseen MWEs in a cross-lingual setting. This can be particularly useful when low-resource languages are involved. We can use training data from high-resource languages to learn information about idiomaticity and use it to identify and classify expressions in low-resource languages. The second research question is *Can data from other languages be leveraged to improve performance of idiomaticity detection?* The results from chapter 5 show also that additional training data from other languages can be leveraged to improve model performance. The *all* setting in Table 5.2 shows that concatenating data from other languages led to an increase in model performance over the monolingual setting i.e., training and testing on the same language. This

can be useful for improving the performance of monolingual systems by incorporating additional training data from other languages.

In our current study, we use two tasks to answer our research questions. Task 1 is the SemEval 2022 task 2 subtask A [102] which is a binary sentence-level classification task of whether a sentence containing a potentially-idiomatic expression includes an idiomatic or literal usage of that expression. In this subtask, the training data consists of English and Portuguese, while the model is evaluated on English, Portuguese, and Galician. As such, the shared task considered evaluation on Galician, which was not observed during training. In this study, we examine cross-lingual settings further, conducting experiments which limit the training data to one of English or Portuguese, to further assess the cross-lingual capabilities of models for idiomaticity prediction.

Task 2 is the PARSEME 1.2 shared task which is a sequence labelling task in which tokens which occur in verbal MWEs, and the corresponding categories of those MWEs (e.g., light-verb construction, verb-particle construction), are identified [78]. This shared task considered a monolingual experimental setup for fourteen languages; separate models were trained and tested on each language. In this study, we consider two different experimental setups: a multilingual setting in which a model is trained on the concatenation of all languages, and a cross-lingual setting in which, for each language, a model is trained on training data from all other languages, and is then tested on the language that was held out during training. We use transformer-based multilingual language models such as mBERT [24], XLM-RoBERTa [21] and mDeBERTa [37].

The contributions of this thesis are highlighted as follows:

1. Proposed novel cross-lingual setups for the PARSEME and SemEval tasks.
2. Showed that models in a cross-lingual setting outperform baselines, demonstrating that models for predicting idiomaticity can generalize to MWEs in

languages that were unseen during training.

3. Demonstrated that models can leverage additional training data from other languages to improve model performance by concatenating the training data in the other languages. This improves over initial monolingual settings.

The remainder of the thesis is structured as follows. Chapter 2 presents the background of research in language models and MWEs in detail. Chapter 3 describes the models used in both of the tasks. Chapter 4 details the datasets used, experimental setup, implementation and parameter settings, and finally the evaluation metrics. Chapter 5 discusses the results obtained from each of the tasks. Chapter 6 outlines the conclusions and future work.

Chapter 2

Related Work

This chapter reviews previous research done on word embeddings including on non-contextual word embeddings and models related to them. It also explores the transformer architecture and transformer-based models such as BERT and its variants. It then explores research done on cross-lingual prediction. It then continues to research done on multiword expressions as a linguistic phenomenon and emphasizing token-level multiword expression identification.

2.1 Word embeddings

Word embeddings are dense, distributed representation of words [2]. They are based on the distributional hypothesis, which states that words are similar to one another if they have similar context around them. They are of two types: count-based word embeddings and prediction-based word embeddings as defined in [2]. These embeddings are fixed length vectors so that various mathematical operations such as addition or average of the vectors could be performed on them to extract relevant information. Previous methods for representing words such as one-hot encoding or TF-IDF created sparse vectors, where the size of the vectors depended on the number of unique words in the corpus on which the embeddings are based, which would result

counts	I	like	enjoy	deep	learning	NLP	flying	.
I	0	2	1	0	0	0	0	0
like	2	0	0	1	0	1	0	0
enjoy	1	0	0	0	0	0	1	0
deep	0	1	0	0	1	0	0	0
learning	0	0	0	1	0	0	0	1
NLP	0	1	0	0	0	0	0	1
flying	0	0	1	0	0	0	0	1
.	0	0	0	0	1	1	1	0

Figure 2.1: Example of a co-occurrence matrix [26].

in very large vectors. To overcome this problem of large vectors, and also better capture semantic and syntactic information about words, word embeddings which were dense representations were introduced. Modern word embeddings, such as the ones created from word2vec [58], often have a vector size of around 300 dimensions.

2.1.1 Count-based word embeddings

Count-based word embeddings, also known as frequency-based word embeddings, are an easy way to represent a word based on the count of each word in a fixed vocabulary. This type of word embedding relies on the use of statistical measures rather than a neural-network based approach. A common method to represent count-based word embeddings is a co-occurrence matrix. A co-occurrence matrix is a matrix X where each entry in the matrix, X_{ij} , is the number of times the word j occurs in the context of the word i [65]. Figure 2.1 shows an Example of how a word-word co-occurrence matrix is formed with a window size of 1 from a very small corpus of three sentences “*I Like deep learning*”, “*I like NLP*” and “*I enjoy flying*” which applies singular value decomposition to a term-document co-occurrence matrix.

A drawback to using a word-word occurrence matrix is due to its sparse nature. If the Vocabulary V of a dataset is in the millions, then the matrix would have many zero elements making it difficult to process in order to create word embeddings. In order to use global statistical information about words, but also reduce the number of non-zero elements, [69] introduced a method known as Global Vectors for Word Representations (GloVe).

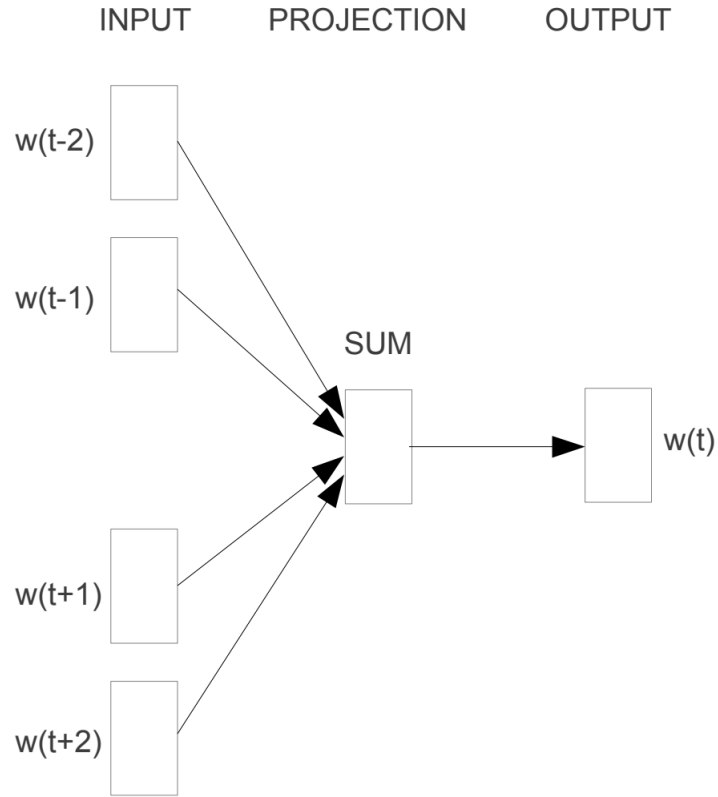
GloVe is trained on the nonzero elements of the word word co-occurrence matrix by calculating the log probabilities of the co-occurrences. The ratio of probabilities between co-occurrences gives us meaning to whether these probabilities contain any semantic information. GloVe is noted to perform better than the popular word2vec at word similarity tasks [69] and with less training time as well. In the WordSim-535 word similarity task [29], GloVe outperforms continuous bag of words (CBOW) model of word2vec [58] and in the rare-word similarity task [57], it outperforms CBOW. In terms of capturing information about MWEs, [45] proved that GloVe is effective in capturing latent semantic information about MWEs using the DiMSUM [92] dataset.

However, [7] showed that prediction-based models such as CBOW perform better on semantic based tasks. In an in-depth analysis between word2vec and GloVe in capturing knowledge regarding semantic compositionality of MWEs, [72] showed that prediction-based word embeddings were better than count-based word embeddings.

2.1.2 Prediction-based word embeddings

Prediction-based word embeddings are created similarly to how neural network models are trained. This type of word embedding is created by assigning each word to a dense vector of fixed size, usually from 50 to 600 [58]. These embeddings are calculated based on the distribution of words in the training corpus. The first use of neural networks to represent words was developed by [9], who created the first Neu-

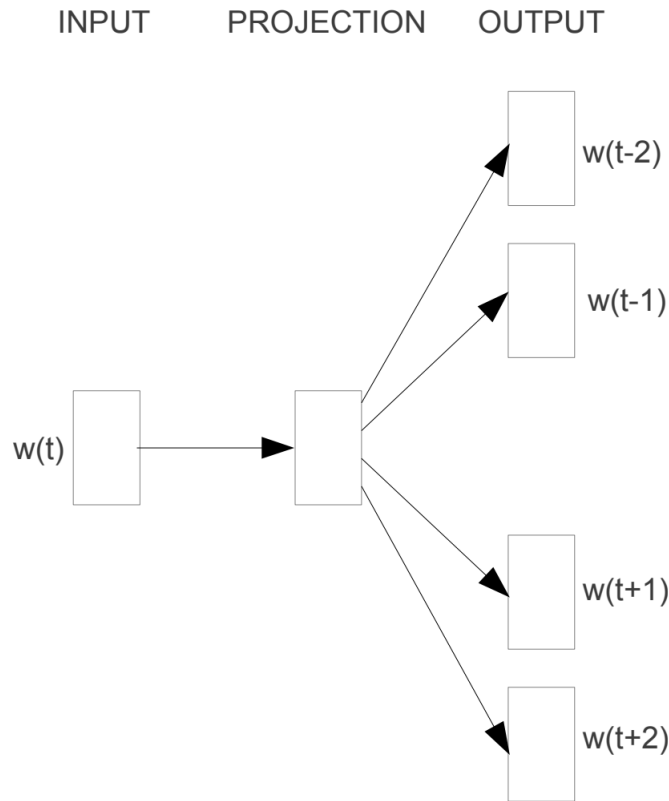
ral Network Language Model (NNLM) which could process and represent text in a large scale. This model had a feed-forward neural network with a non-linear hidden layer used to predict words. Although this was a language model used to predict words, the main objective was not to learn word embeddings, but embeddings were a by-product of it. The first model specifically created for learning word embeddings was [20], where to create a word embedding for a given word in the training corpus, it uses the context of that particular word. For larger datasets containing billions of words, and to process these words at a faster rate, [58] introduced word2vec. It showed improvements in creating word embeddings compared to previous models [58], while at the same time showing that it can capture syntactic and semantic information about words. Word2vec initially was created with two different architecture. First was the continuous bag-of-words (CBOW), where the word embedding for the current word $w(t)$ is predicted based on its context as shown in Figure 2.2.



CBOW

Figure 2.2: Architecture of CBOW [58].

CBOW is similar to the model developed by [9], except with the absence of the non-linear hidden layer and in this case all words around the target word are projected onto a single vector by averaging the vectors of the words around it. The second word2vec approach was the continuous skip-gram model which is similar to CBOW, but here it tries to predict the context for a target word as shown in Figure 2.3. Here, the model tries to predict the words in the context C around the target word $w(t)$ trying to predict words in C . As the size of the context increases, the accuracy of word2vec increases. However, a higher range C required more computing power. In Figure 2.2 and Figure 2.3, $C=2$ and the context words are $w(t-2)$, $w(t-1)$, $w(t+1)$ and $w(t+2)$. The authors use $C=10$ in their initial experiments. Both of these



Skip-gram

Figure 2.3: Architecture of skip-gram [58].

approaches use a hierarchical softmax layer [63, 64] to calculate the probabilities, which is an efficient way to compute the softmax function.

To improve skip-gram, [61] introduced skip-gram with negative sampling (SGNS). While skip-gram tried to predict the context for each target word, the authors introduced a new optimization function known as negative sampling to make the model more scalable and learn faster. Here, in addition to predicting all of the context words which are positive examples, it also selects random words as negative samples. It tries to identify which words are positive, i.e., words in context, and which are words are negative, i.e., words out of context, making it a binary classification task.

This led to a small number of weights being updated, thereby decreasing computation time. In the word analogy task defined in [58], SGNS has the best accuracy among word2vec models. Overall, word2vec has shown to be an improvement over count-based models such as GloVe in various NLP tasks, such as in topic segmentation [66]. It also works better monolingually across different languages in the same task[66].

While [58, 61, 62] succeeded in creating better and more efficient representations of words, they also showed that these models can capture semantic information about them. Figure 2.4 shows how neural word embeddings created from models such as word2vec can capture semantic information. We can see that the distance between “Queen” and “King” is similar to the distance between “Man” and “Woman” in the graph depicted in the right side of Figure 2.4. Also, we can see the red vectors in the graph depicted in the left side of Figure 2.4 shows that size of the red vectors are roughly the same size. Word2vec can also find relationships between pairs of words. For example, we can calculate the vector for “Queen” using the following equation from Figure 2.4 :

$$vector(“King”) - vector(“Man”) + vector(“Woman”) = vector(“Queen”)[58]$$

While these models provided dense, efficient representation of words, they had drawbacks. Firstly, for languages such as Finnish, Arabic and Japanese which are morphologically rich in nature and contain many unique wordforms, representing words in these languages would be difficult using the above mentioned methods. Also, models such as GloVe and word2vec cannot handle out-of-vocabulary (OOV) words. These are words that do not appear in the training dataset but occur in the test dataset. To overcome this, fastText [12] was introduced. While word2vec and GloVe create an embedding for each word in the vocabulary, fastText creates character n-grams and each word is represented by these character n-grams. Character n-grams

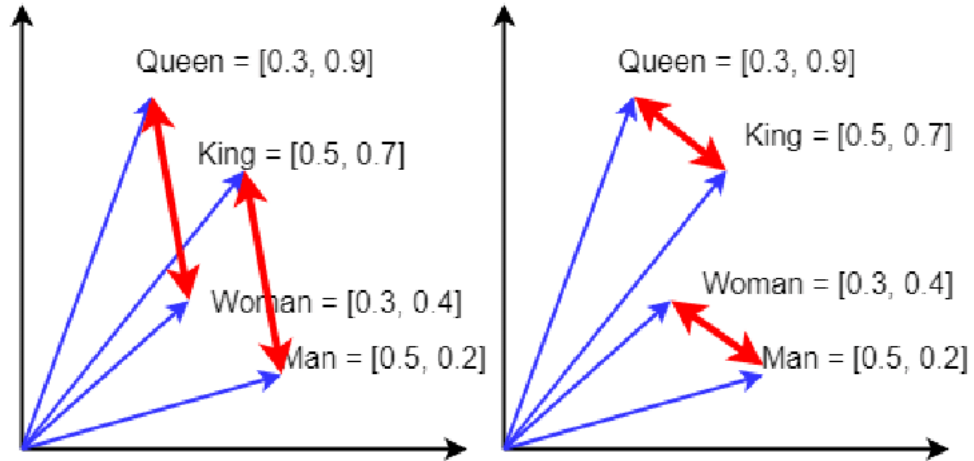


Figure 2.4: Illustrated example of word embeddings of words “King”, “Queen”, “Man” and “Woman” projected in a 2D space [98].

are subwords which are of length less than or equal to n , where n usually ranges between three and six. E.g., for *where*, the subwords created for $n=3$ are [12] :

$$\langle wh, whe, her, ere, re \rangle$$

FastText creates a unique embedding for each subword and then averages the embeddings of all the subword embeddings to form the embedding for the whole word. It outperforms previous word2vec models in various downstream NLP tasks such as word similarity and word analogy tasks across a range of languages [12].

Prediction-based word embedding models create a single global vector for each word. This can be a problem when a single word is used in multiple ways. For example, in the sentence *My friend works in a bank*, the word *bank* indicates a financial institution. It can also be used to indicate the edge of a river where it meets the land in the following sentence *I was fishing by the river bank*. The same word *bank* has different meanings based on its context. To represent the meaning of a word based on its context, contextual embeddings were proposed. One of the first context embedding language models that was proposed was TagLM [70]. This model

created bidirectional word representations using bidirectional Recurrent Neural Networks. The state of the art contextual embedding model is Bidirectional Encoder Representations from Transformers, also known as BERT [24].

2.2 BERT

BERT [24], developed in 2018, is a revolutionary language model that could perform multiple downstream tasks. BERT is trained like a multi-purpose language model, where it needs to be pre-trained only once on a large unannotated corpus, and for each downstream task, it just needs to be fine-tuned for that task. BERT is based on the transformer architecture, proposed by [106]. Transformer improves over existing models such as RNN (recurrent neural network) or LSTM (long short-term memory). Before we discuss about BERT, we need to explain briefly about RNN-based language models and the underlying transformer-based architecture.

2.2.1 RNN

RNN, proposed by [82], allowed models to process data sequentially. It is especially useful in NLP, by learning contextual information from words it has already processed. The first RNN-based language model was proposed by [59]. The architecture of the model is illustrated in Figure 2.5.

The middle layer of the model is called the hidden layer or the context layer. Each input word is represented as a vector of fixed length, similar to how word embeddings were created using word2vec. In the RNN-based language model, the input vector to $context(t)$ is calculated by concatenating the input vector of the current word $input(t)$ and the outputs of the context layer $context(t-1)$, which are the weights of the hidden layer after the previous word was processed.

RNN improved over n-gram language models in various speech recognition tasks [59].

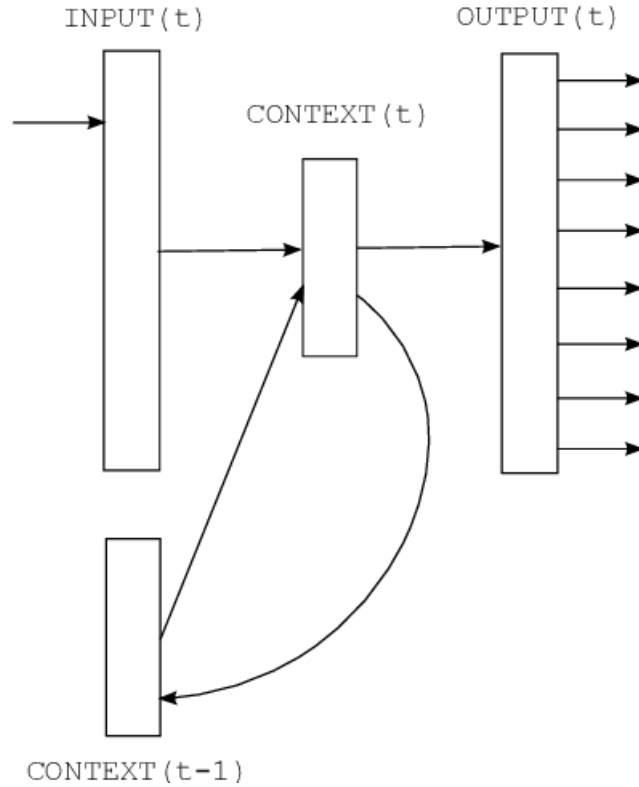


Figure 2.5: Architecture of the RNN-based language model [59].

A major drawback to the RNN-based models was the *vanishing gradient* problem [39], where as the input grows in size, the gradient becomes too small to update the hidden state of the network. This would result in the neural network not being able to capture long range dependencies and losing important information. To address this problem, long-short term memory networks (LSTMs) [40] were proposed, which introduced a gating mechanism known as the ‘forget’ gate, which decided how much of the data from the previous time step needs to be ‘forgotten’ and what information needs to be remembered, allowing only important information to be processed.

[15] showed that longer sentences led to a decrease in performance by RNNs. To understand context in larger sentences, a mechanism known as “attention” was proposed [4] for RNNs. The attention mechanism, allows the model to concentrate on (i.e., attend to) the important parts of the input based on their relevance to the word the decoder is currently processing. Figure 2.6 shows the working of the

attention mechanism. For a given target word y_t , the input sequence is denoted by $x_1, x_2, x_3, \dots, x_T$, s_t is the hidden state of the RNN at time t . $h_1, h_2, h_3, \dots, h_T$ indicates a sequence of annotations, to which the encoder maps the input sequence [4]. $a_{t,1}, a_{t,2}, a_{t,3}, \dots, a_{t,T}$ is the alignment model, which tells the model which parts of the input sequence are important. The model is designed as a feed-forward neural network where the input is processed sequentially. The sum of the alignment model creates the context vector, which is then concatenated with the output of the decoder at the previous step to create the output at the current step.

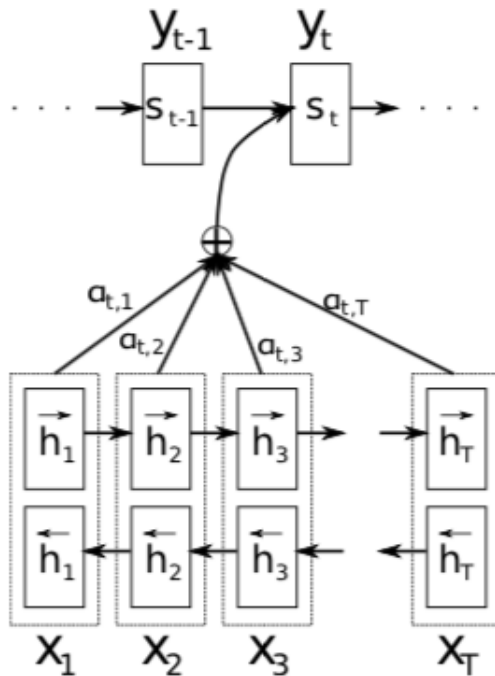


Figure 2.6: Working diagram of the attention mechanism [4].

The use of attention mechanism has shown to improve performance in tasks such as machine translation [4].

2.2.2 Transformers

Transformers [106], improved over RNN and LSTM-based language models. Due to the nature of the recurrent models, to process the input sequentially, this would

create a sequence of multiple hidden states h_t , based on the previous hidden state h_{t-1} , for a word at position t . The bigger the dataset, the larger the number of hidden states created. This would result in memory constraint issues when dealing with larger datasets. To address this memory issue, and also to allow parallelization of the training process to reduce training time, transformers was proposed. It uses multiple encoder blocks to represent the input and multiple decoder blocks to produce the output. The encoder creates representations of the input which also captures contextual information for each word in the input. It captures the information by creating a context vector of fixed length. The decoder then produces the output sequence using the hidden state of the encoder block. Another essential part of transformers is the attention mechanism.

Transformers use a unique form of attention, known as “self-attention” or “intra-attention”, proposed by [106]. While the traditional attention mechanism uses the entire input sequence to calculate the hidden states, self-attention also deals with different positions related within the sequence itself, allowing to capture dependencies between the words in a sequence. Figure 2.7 shows the working of the transformer architecture. Transformers contain blocks of attention layers known as multi-head attention. These attention blocks help the model to capture the above mentioned dependencies in the input sequence. For a given input sequence, the attention blocks find the relevant words for each word in the input sequence by considering the entire sequence as a whole. The encoder uses layers of self-attention stacked on itself to process the inputs and similarly the decoder uses layers of self-attention stacked on itself to generate the output. Given an input sequence $x_1, x_2 \dots x_n$, the encoder creates continuous representations $z_1, z_2 \dots z_n$. This passes as input to the decoder layer. The decoder creates an autoregressive [33] output $y_1, y_2 \dots y_n$, where the output generated at any time is dependent on the previously generated word [106].

[106] showed that transformers improved over neural networks models in tasks such

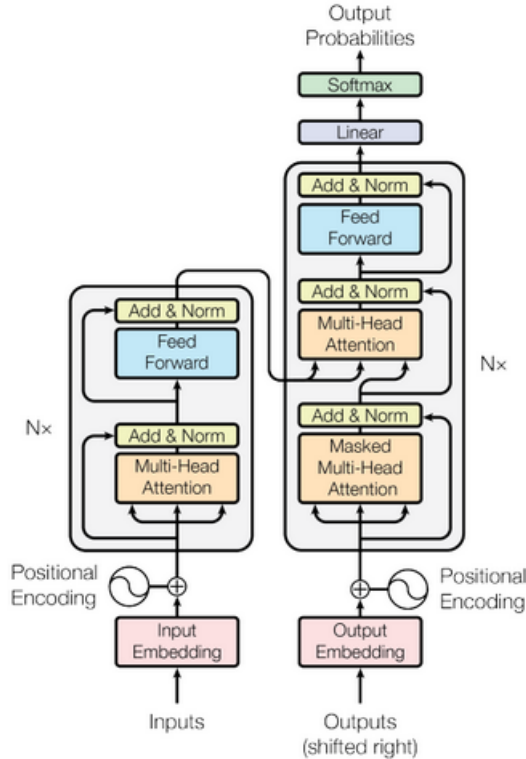


Figure 2.7: Model of the transformer architecture [106].

as machine translation. It also gives improvements in tasks such as text classification [114], image recognition [68] and question answering [25]. Some transformer-based language models are BERT [24], generative pre-trained transformer (GPT) [73], XLNET [114], T5 [75] and BART [51]. Since our current study focuses on the use of BERT and BERT-like models, we give a detailed description of BERT.

2.2.3 BERT architecture

BERT is unique from other transformer-based language models. It contains only the encoder part of the transformer architecture. By utilising both the right and left context of the input sentence, it creates bidirectional representations of words. Unlike GPT, which only does left to right training [73], BERT does bidirectional training, allowing it to not only capture information about words occurring before a

target word, but to also capture information about words that occurs after it. This is done through the use of masked language modeling (MLM).

For input tokenization, BERT uses wordPiece embeddings [112], where subwords are created to handle rare and OOV words. Figure 2.8 illustrates the input representation of BERT. A single input sequence consists of a sentence pair, which is denoted by the segment embeddings, where A corresponds to the first sentence and B corresponds to the second sentence. There are special input tokens which are the [CLS] token and [SEP] token. The [SEP] token is used to differentiate the sentences while the [CLS] token is a token that contains the overall information about the sequence. Each input word of the sequence is represented as a summation of its corresponding position, segment and token embedding. These input tokens can be seen in the top input layer of Figure 2.8. For each sentence pair, the model produces a classification token [CLS], which is used as a representation of the sentence for classification tasks. It also contains a separator token [SEP], that can be used to differentiate between sentences such as questions and answers.

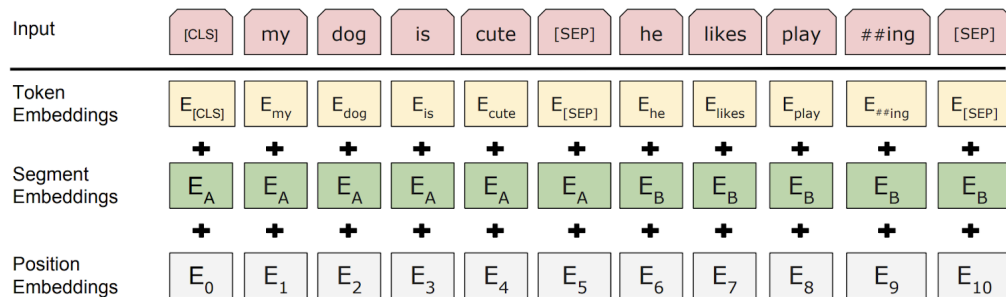


Figure 2.8: BERT input representation [24].

The usage of BERT occurs in two stages: 1. pre-training of the model and 2. fine-tuning of the model. This is highlighted in Figure 2.9. The left side of Figure 2.9 illustrates the pre-training stage of BERT, while the right side shows the fine-tuning stage. BERT is pre-trained on English Wikipedia which contains about 2.5 billion

words, and the BooksCorpus dataset [117], which contains around 800 million words. It is pre-trained by using two tasks. The first is masked language modelling (MLM), where the model randomly masks 15% of the input sequence with a mask token and the model is forced to predict these tokens. The masked word is passed through an output softmax layer where the word with highest probability is the predicted word. Out of this 15% of tokens, 80% are replaced with a [MASK] token, 10% are replaced with a random word and the remaining 10% do not change. This allows to avoid creating a mismatch between the pre-training and fine-tuning stages. The second pre-training task is next sentence prediction (NSP). In this task, the model has two sentences A and B as input taken from the corpus. In 50% of the training pairs, the sentence B is the next sentence after A in the corpus. In the other 50%, B is a random sentence from the corpus. Given these pairs, the model has to predict whether sentence B actually appears after A . This task is useful in downstream tasks such as question answering (QA) [24].

The second stage of BERT is the fine-tuning stage. For a specific task, we use the existing pre-trained parameters of the model and fine-tune with the task-specific data. This is highlighted in Figure 2.9, where the pre-trained model is fine-tuned for three different tasks which are SQuAD [76], name-entity recognition (NER) [104] and multi-genre natural language inference (MNLI) [109].

BERT had two model sizes denoted by $BERT_{BASE}$ and $BERT_{LARGE}$. $BERT_{BASE}$ had (L=12,H=768) and $BERT_{LARGE}$ had (L=24,H=1024), where L denoted the number of self-attention layers and H denoted the hidden layer size, i.e., the number of dimensions per token. $BERT_{LARGE}$ achieved state-of-the-art results in various NLP tasks such as SQuAD [76], NER [104] and GLUE [107]. It improved over $BERT_{BASE}$ and LSTM models, while reducing the training time as fine-tuning is inexpensive over training from scratch. To handle tasks in other languages, multi-

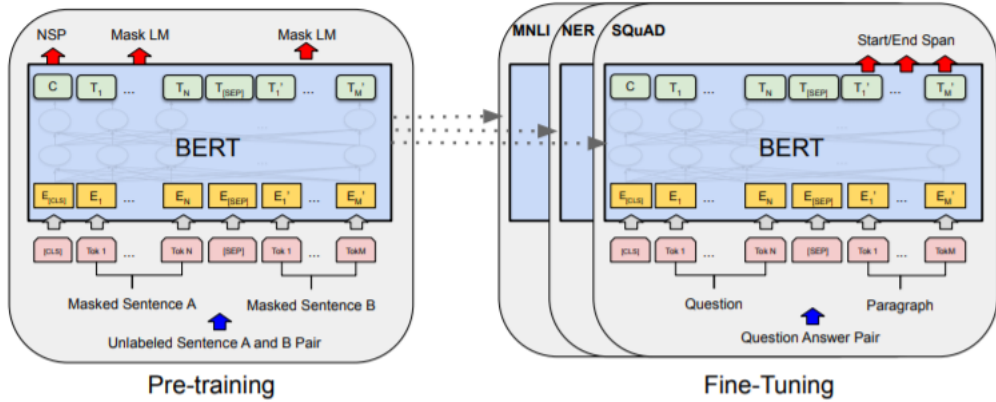


Figure 2.9: Pre-training and fine-tuning stages of the BERT framework [24]. The architecture is similar in both stages, except for the output layers. The output tokens used in the fine-tuning stage depend on the task it is being fine-tuned for.

lingual BERT (mBERT) was introduced which is pre-trained on 104 languages.¹ Robustly Optimised Pre-training Approach (RoBERTa) [56] was introduced as an improvement over BERT. It contained a few key changes from BERT. Firstly, it was pre-trained on a much larger dataset. Secondly, it does not use the NSP objective while pre-training. Thirdly, it uses Byte-Pair Encoding (BPE) [95], similar to the one used in GPT-2 [74] as its tokenization method rather than wordPiece. Lastly, it uses a different masking method in the MLM part of pre-training. BERT uses *static masking*, where the masking is done only once while pre-training. In contrast, RoBERTa uses *dynamic masking*. In this method the dataset was duplicated multiple times to ensure different words get masked each time since the masking is done randomly. [56] showed that the use of *dynamic masking* resulted in improved performance of the model in the SQuAD and MNLi tasks.

In our current study, we use XLM-RoBERTa [21], which is a multilingual version of RoBERTa. [21] mentions a trade-off between model performance and the number of languages the model is pre-trained known as the *curse of multilinguality* and suggests that the way to solve this problem is to increase model size. Section 4.3 gives a

¹<https://github.com/google-research/bert/blob/master/multilingual.md#list-of-languages>

more detailed description on XLM-RoBERTa’s model parameters. XLM-RoBERTa outperforms mBERT in various cross-lingual NLP tasks such as the Cross-lingual Natural Language Inference (XNLI) [22] and the Multilingual Question Answering (MLQA) [52].

Decoding-enhanced BERT with disentangled attention (DeBERTa) [38] improved upon BERT and RoBERTa by introducing an attention mechanism known as *disentangled attention*. Unlike BERT and RoBERTa which has a single vector for each word which is the sum of the word embedding, segment embedding and position embedding (Figure 2.8), DeBERTa has two vectors for each word. The first is the word embedding and the second is the positional embedding of the word. The attention weights are then calculated using these two vectors for each word. DeBERTa also introduced an *enhanced attention decoder* where during MLM pre-training, the model also introduces the absolute positional embedding of the masked word before passing it through the softmax layer. These changes led to an improvement over BERT and RoBERTa in the MNLI and SQuAD tasks [38].

In our current study, we use a multilingual version of DeBERTa known as mDeBERTa [37], which is based on the DeBERTa V3 model [37]. In DeBERTa V3, the MLM pre-training task is replaced with Replaced Token Detection (RTD) [19]. RTD is a pre-training objective that is similar to a generative adversarial network (GAN) [32] in that a generator generates words while a discriminator needs to distinguish the generated words from the original words. XLM-RoBERTa and mDeBERTa is pre-trained on the same CC100 corpus [21]. mDeBERTa improves over XLM-RoBERTa in the XNLI test [37].

2.3 Cross-lingual

The term “cross-lingual” in NLP is closely related to the term “multilingual”. Multilingual refers to when a certain task or model involves the use of multiple languages. It generally occurs when the task has a training dataset in multiple languages and is tested on the same set of languages. Cross-lingual transfer is a more narrow work within NLP and occurs when a model or task has a training dataset in one or more languages and is tested on a separate set of languages. One common NLP task that is related to cross-lingual work is machine translation. Machine translation (MT) is the task of automatically translating text from one language to another. The use of neural-network models to perform machine translation is known as neural machine translation (NMT). [42] proposed the use of recurrent neural models for MT. [99] improved upon this by the use of sequence to sequence models for MT by taking entire sentences for input and converting them word by word. [99] also used a single neural network instead of multiple models connected with one another. Other work in NMT involves the use of a character-level encoder model [53] and character-level decoder model [17]. While much of the work has been small-scale theoretical models, MT has been a major area of research in industry. [112] showed how Google’s NMT system works and its working at scale through the use of the encoder-decoder model and the attention mechanism. Transformer-based models showed great improvement in NMT while reducing training time through pre-training. [49] showed that the use of pre-trained transformer-based models such as XLM improved over recurrent models. This was done through the use of translation language modelling (TLM). TLM involves the masking of tokens in both the training and test languages while pre-training. This is illustrated in Figure 2.10.

In addition to predicting masked tokens in English, the model is also forced to predict masked tokens in French while pre-training. The English and French words have the same position embeddings. [21] showed that XLM-RoBERTa outperformed

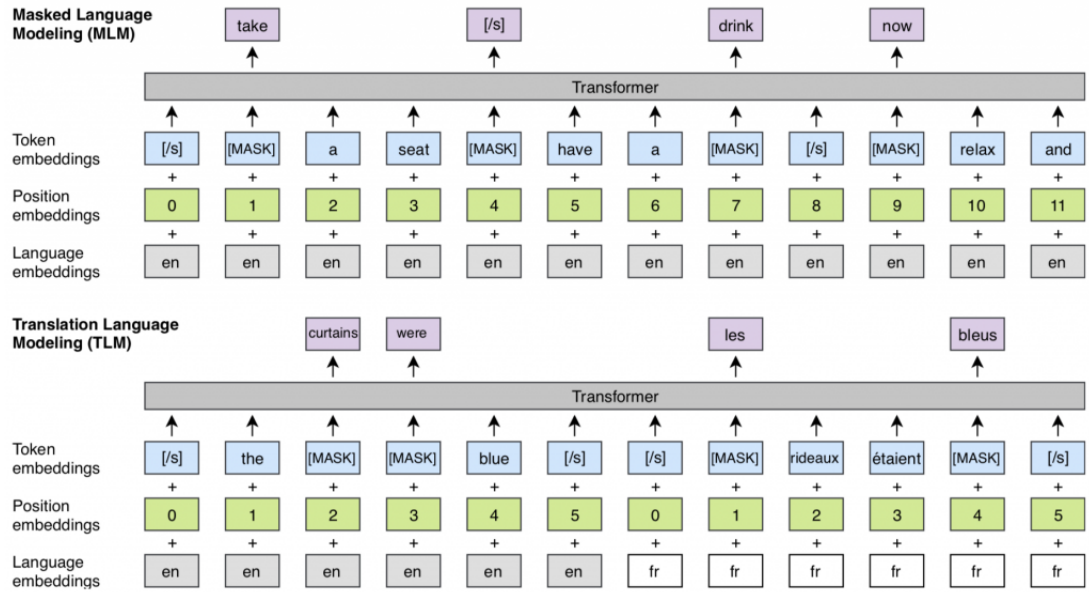


Figure 2.10: Masked language modelling vs translation language modelling [49].

XLM and mBERT in various cross-lingual tasks such as XNLI MT task and MLQA. Other cross-lingual tasks include cross-lingual document classification [94] and cross-lingual information retrieval (CLIR) [41]. Cross-lingual document classification was first introduced by [8]. It involved having training documents in one language, and classifying documents in another language. Cross-lingual information retrieval refers to the task where given a query in one language, the model has to retrieve documents in a different language relevant to that query [8].

While the majority of the cross-lingual work has been on high-resource languages, such as English, German, French, etc., little work has been done in low-resource languages. *Cross-lingual transfer* allowed models to learn to *transfer* information from one language to another. This can be done through the use of cross-lingual word embeddings [81]. Cross-lingual embeddings are representations of words where the translations of words appear closer to one another when projected on a joint embedding space. This is illustrated in Figure 2.11. The pink words are words in Italian and the green words are its English translations.

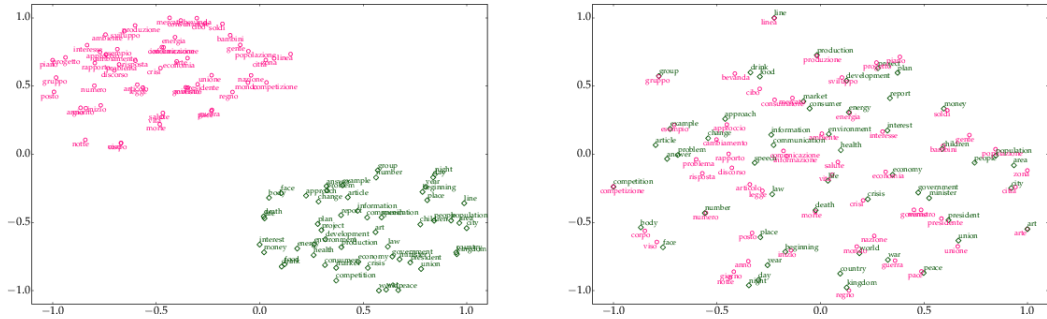


Figure 2.11: monolingual word embeddings (left) and cross-lingual word embeddings (right) [81].

[60] proposed a method that uses linear transformation of words across languages to learn cross-lingual word embeddings. [35] proposed a method to learn cross-lingual word embeddings for low-resource languages from a high-resource languages and bilingual lexicon. [111] showed that pre-trained transformer based models, such as mBERT, are able to create deep contextualized cross-lingual word embeddings since they have a joint vocabulary for all the languages it was pre-trained on.

2.4 Multiword Expressions

This section explains the work that has been done in multiword expressions in the context of NLP. Multiword expressions (MWEs) are a frequent phenomenon that occur in natural languages. They are described as expressions containing multiple words, where the meaning of the expression cannot be derived from its component words i.e., they display some form of idiomaticity [84]. In this section we focus mainly on two NLP tasks associated with MWEs, which are *token-level idiomaticity prediction* and *token-level MWE identification*. Before we discuss these areas, we give a brief introduction as to the importance of MWEs in NLP and challenges language models face when processing them.

There are multiple types of MWEs that exist. One common type of MWE is noun compounds (NCs) where two or more nouns combine to form a MWE (eg., *red*

flag). [5] first proposed to predict the compositionality of a MWE focusing on noun compounds using latent semantic analysis (LSA). This was later extended by [46], who proposed an unsupervised system to predict the compositionality of noun compounds. [79] introduced a dataset that not only analyses the noun compound as a whole but also determines how much each component of the noun compound contributes to the meaning. Most research in noun compounds predicted them at a type level, but it is also imperative we predict them on a token-level. Type-level idiomaticity refers to when we predict the idiomaticity of the entire MWE without looking at the component words. Token-level idiomaticity refers to when we predict how much individual tokens in a MWE contributes to the idiomaticity of the entire expression. [30] showed that predicting noun compounds at a token level also improves the model’s ability to predict the compositionality of noun compounds.

Another common type of MWEs is verb-particle constructions (VPCs). VPCs are made of a verb and a particle which can be either a preposition (eg., *play around*), an adjective (eg., *cut short*) or a verb (eg., *let go*) [6]. [11] proposed a method to identify VPCs that may not appear in the dataset by largely identifying them as compositional in nature.

Due to the potential idiomatic nature of MWEs, MWEs are hard to identify [87]. MWEs are important to identify and extract due to the growing number of MWE instances [28, 77, 78, 102]. For a large number of downstream tasks, it is important to automatically identify MWEs to improve model performance. One such task is machine translation. We can leverage bilingual MWEs for such tasks. [80] define bilingual MWEs as MWEs which have a one-to-one word-for-word translation from a source language into a target language. The use of bilingual MWEs have been shown to improve machine translation performance [80]. [115] proposed a biLSTM based model to translate MWEs from a source language to MWEs in a target language. They also introduced a new metric known as mwe_{score} to evaluate the effectiveness

of the translated MWE compared to the original meaning of the source MWE.

Another NLP task where MWE identification is important is information retrieval (IR). Information retrieval is defined as the task of identifying relevant information from a large corpus given a query. [50] created a manually annotated French corpus of MWEs that help with IR. [1] demonstrated that treating MWEs as a single unit compared to analysing the individual tokens of the MWEs improved the performance of the model. For example, the word *pop star* means a celebrity, and if the query contains the words *pop star*, it is necessary to treat them as a single expression to identify relevant information from the corpus.

A lot of work has been done in token-level idiomaticity prediction. This task involves predicting whether a given MWE is idiomatic in nature or not, while examining it at a token-level. [43] proposed and evaluated multiple word-embedding based models such as word2vec and CBOW for predicting the idiomaticity of verb-noun combinations (VNCs). They also showed that averaging the word embeddings of the MWEs proved effective in identifying the semantic nature of the expression. Another way for identifying the idiomaticity of an MWE is by examining the context around it. This can be done through the use of contextual embeddings. Contextual embedding models such as ELMo [71] and BERT [24] have been shown to improve performance in predicting idiomaticity in MWEs over non-contextual models such as word2vec [36]. A task that is similar to our current study is [27]. It involves the use of BERT-like models to predict the idiomaticity of MWEs in a cross-lingual setup, while also extending to expressions that are unseen, i.e., in a cross-lingual “zero-shot” setting. Here, the MWEs in the test dataset do not appear in the training dataset. While zero-shot settings can also be monolingual, [27] performs a cross-lingual zero-shot task. They train in English and test in Russian and vice-versa, making it a cross-lingual task. In our current study, we use the dataset presented in “SemEval Task 2: Multilingual idiomaticity prediction detection and sentence embedding”

[102]. This is a shared task where multiple teams present their solutions to a given problem. This task is a binary classification task of predicting the idiomaticity of an expression as idiomatic or literal in nature. Most of the solutions presented were transformer-based models [113, 16, 103], which were the best performing solutions at the shared tasks.

Another major area of research in MWEs is the identification of MWEs. This is where models have to identify MWEs in running text. The first major shared task in labeling MWEs was the SemEval 2016 task 10 titled “Detecting Minimal Semantic Units and their Meanings ” (DiMSUM) [92]. The dataset consisted mainly of social media text which was pre-processed. Given a sentence, the model has to identify which words are parts of MWEs. Most of the systems presented heuristics based solutions or conditional random field (CRF) based solutions except for [90] which used a simple neural network perceptron based solution. [31] proposed the first deep-learning based model for token-level classification of MWEs. They used a recurrent neural network and convolutional neural network (CNN) on the DiMSUM dataset. They improved over the existing solutions presented in the initial shared task with the use of the CNNs. While most token-level based tasks focused on identification of other types of MWEs, little work has been on verbal MWEs (VMWEs). VMWEs are MWEs that contain a verb as a part of the MWE. For example, the expression *make a meal* contains the verb *make* [88]. It has two meanings, the first is an idiomatic sense which means to spend more time and effort on a task than is actually required. The second meaning is the literal meaning which is to make food. [88] created the first edition of the PARSEME shared task (PARSEME 1.0), which focused on the automatic identification of VMWEs in multiple languages. The first edition includes VMWEs in 18 languages. The second edition of the PARSEME shared task (PARSEME 1.1) [77] introduced major changes such as new languages including Basque, Croatian, English and Hindi. It also refined the meaning of VMWE as an

expression where the head of the expression is a verb. In our current study, we use edition 1.2 of the PARSEME shared task [78]. Compared to the previous editions of the PARSEME shared task, Chinese, Irish and Swedish were newly introduced or significantly changed in PARSEME 1.2. This edition also emphasized the evaluation of unseen MWEs and also redefined the meaning of unseen VMWEs as the multiset of lemmas that are not annotated in either of the training or dev datasets, as opposed to only in the train dataset as defined in earlier PARSEME editions. In this shared task, MTLB-STRUCT [100] is the best performing system. MTLB-STRUCT uses mBERT in conjunction with a dependency parser to identify and classify VMWEs. The architecture of MTLB-STRUCT is illustrated in Figure 2.12.

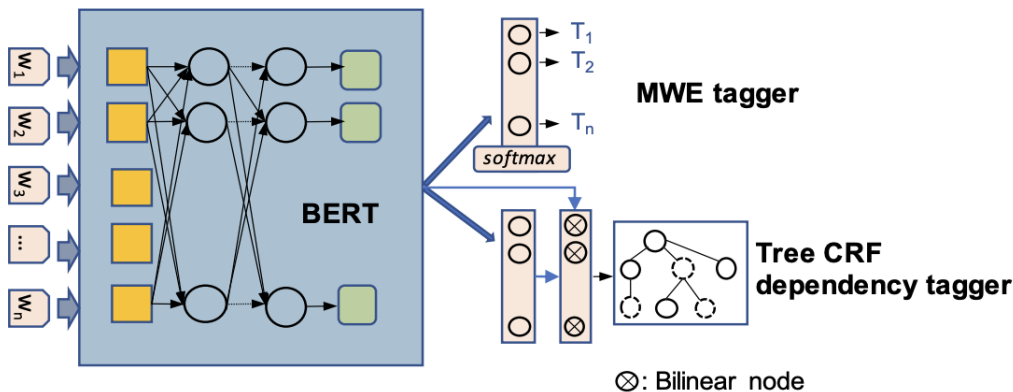


Figure 2.12: Architecture of MTLB-STRUCT [100]. The left side of the image contains the input to the BERT model used. The right side contains two different parts which are the layers on top of BERT. The first is a linear layer used for MWE tagging and the second is Tree CRF used to improve the performance of the model.

A detailed explanation of MTLB-STRUCT can be found in 3.2. The model performs the best across all the languages in the shared task hence we decide to use this system in our experiments.

Chapter 3

Models

We use transformer-based multilingual BERT models since our experiments involve training and testing in multiple languages.

3.1 SemEval

For SemEval 2022 task 2 subtask A [102] we apply BERT [24] models for sequence classification. In the initial shared task published in 2020, the shared task organizers used a multilingual BERT (mBERT) model for the baseline [102]. We follow the baseline where the model is fine-tuned on the training data and is tested on the testing data as usual. We wanted to test if more-powerful models improve the classification performance. To do this, we extend the baseline to use more-powerful multilingual models, including XLM-RoBERTa [21] and mDeBERTa [37], as opposed to mBERT.

3.2 PARSEME

For the PARSEME 1.2 task [78], we use the MTLB-STRUCT system [100], which performed best overall in the shared task. The architecture of MTLB-STRUCT is

setup in a way such that it simultaneously learns MWEs and dependency trees by creating a dependency tree CRF network [83] while using the same BERT weights for both the tasks. MTLB-STRUCT uses a pre-trained multilingual BERT model. A layer is added on top of the pre-trained model that contains a softmax function which performs classification of MWEs based on their category. The loss obtained during classification of MWEs is optimised using the cross-entropy between the training label and the predicted label. Parallely, for the dependency trees, a layer containing a dependency conditional random field (CRF) performs prediction for dependency trees. A separate loss for dependency trees is calculated and then both losses are added which is then optimized by the ADAM optimiser [44]. The use of this architecture has been shown to improve the cross-lingual identification and classification of VMWEs as shown in [101], where the model was trained on the German dataset in the PARSEME 1.1 [77] shared task and tested on English data from the same shared task. We evaluate this system in novel cross-lingual and multilingual experimental setups as explained in section 4.2.

We chose these systems because we wanted to find a system where we could reproduce the results of the original system. These systems performed well in the monolingual setup in the initial shared task and we wanted to expand this to our novel cross-lingual and multilingual setups and see if BERT-based language models can learn information about idiomaticity that is language specific or not.

Chapter 4

Materials and Methods

In this chapter, we describe our datasets, experimental setup, implementation and parameter settings, and evaluation metrics.

4.1 Datasets

In this work, we focus on cross-lingual identification and classification of MWEs. To perform cross-lingual analysis, we need datasets that contain instances in multiple languages. For this, we use the SemEval 2022 task 2 subtask A [102] which is titled “Multilingual Idiomaticity Detection and Sentence Embedding” and the PARSEME 1.2 [78] edition on identification of verbal multiword expressions.

4.1.1 SemEval

SemEval 2022 task 2 subtask A contains a multilingual dataset which is further divided into train, dev, eval, and test sets. We train models on the train set and evaluate on the test set, which was used for the final evaluation in the shared task. While only the train and test datasets are used in the experiments and our results are reported on the test set in Table 5.1, we initially used the dev and eval datasets to test the feasibility of our models. We do this by reproducing the results of the

MWE	Previous	Target	Next	Label
Brain drain	The administration gutted government offices of cybersecurity expertise.	They suffered a serious cybersecurity brain drain.	Learn what you need to know about defending critical infrastructure.	0
relógio analógico	A Casio não faz apenas relógios digitais.	Este é um elegante relógio analógico da marca japonesa.	Outro exemplo é a pandemia de 2009 da Gripe Suína, Com um ótimo custo-benefício, dá para agradar o papai, sem prejudicar seu bolso.	0

Table 4.1: Examples from the SemEval 2022 task 2 subtask A dataset, which are lightly edited to make them more concise.

original baseline to ensure we can extend the baseline to our novel cross-lingual setup. The dataset includes instances in three languages: English (en), Portuguese (pt) and Galician (gl). The dataset contains the following columns: MWE, Previous, Target, Next and Label. The previous column contains the sentence that precedes the sentence that contains the target MWE. The target column contains the sentence which has the target MWE and the next column contains the sentence which follows the target sentence. The label column has values of either ‘0’ or ‘1’ where 0 means the target MWE has been classified as idiomatic and 1 means the target MWE has been classified as literal. The train data contains 2535 idiomatic instances which is about 56.4% of the training set and 1956 literal instances which is about 43.5% of the training set. Table 4.1 shows how the dataset is structured with an example in English and Portuguese. In both the examples, the MWEs are labeled as ‘0’, where the MWEs is used in an idiomatic sense.

We only consider the “zero-shot” setting from the shared task. For this setting,

Language	Train	Test
English	3327	916
Portuguese	1164	713
Galician	0	713

Table 4.2: The number of training and testing instances in the SemEval dataset.

the training data consists of English and Portuguese, while the test data includes these languages and also Galician. In this thesis, we consider further cross-lingual experiments in which a model is evaluated on expressions in a language which was not observed during training. Since the nature of the experiments are cross-lingual, i.e., the training and testing languages are different, they are also zero-shot experiments. Specifically, we explore models that are trained on one of English or Portuguese. We evaluate on the test dataset, and focus on results for languages that were not observed during training (e.g., when training on English, we focus on results for Portuguese and Galician). The train data has 4491 instances in total of which there are 3327 English instances and 1164 Portuguese instances. The test data consists of 916, 713, and 713 English, Portuguese, and Galician instances, respectively. Table 4.2 contains the training and testing split of the SemEval dataset. Since there are no training instances in Galician while there are testing instances in Galician, we can perform cross-lingual experiments where the model is trained on English, Portuguese or both and tested on Galician.

4.1.2 PARSEME

For PARSEME 1.2 [78], the shared task dataset contains sentences with token-level annotations for verbal MWEs (VMWEs). VMWEs are MWEs where, in their prototypical form, the head of the expression is a verb. In this dataset, VMWEs are in fourteen languages which are German (de), Greek (el), Basque (eu), French (fr), Irish (ga), Hebrew (he), Hindi (hi), Italian (it), Polish (pl), Brazilian Portuguese

Output	*	*	*	*	1:VID	*
Input	la	médiocrité	de	l'insonorisation	pose	vraiment
Output	1	*	*	*	*	*
Input	problème	:	télévision	,	discussion	.
Output	*	*	*	*	*	*
Input	.	vous	entendrez	tout	ce	qui
Output	2:IRV	2	*	*	*	
Input	se	passé	chez	vos	voisins	

Table 4.3: Example of an input for the PARSEME task dataset in french, where the sentence has been lightly modified to make it more concise.

(pt), Romanian (ro), Swedish (sv), Turkish (tr), and Chinese (zh). In this task, the model is expected to identify the verbal multiword expression in the test example and also categorize it.

For example, in the given input, *la médiocrité de l'insonorisation pose vraiment problème : télévision, discussion.. vous entendrez tout ce qui se passe chez vos voisins*. Table 4.3 shows the token-level input and the output for this input. * indicates the gold standard has not identified that particular token as a VMWE. In this example, the gold standard indicates that *pose problème* is a verbal idiom (VID) and *se passe* is an inherently reflexive verb (IRV).

Compared to the previous editions of the PARSEME shared task, Chinese, Irish and Swedish were newly introduced or significantly changed in PARSEME 1.2. This edition also emphasized the evaluation on unseen MWEs and also redefined the meaning of unseen VMWEs, to be the multiset of lemmas that are not annotated in the train and dev datasets, as opposed to only in the train dataset as in PARSEME 1.1. Each language dataset contains approximately 300 unseen VMWEs. The data for each language is divided into training, development and testing sets. Table 4.4 contains the training, development and testing split for the PARSEME dataset among languages and an average over all languages.

Language	Train	Dev	Test
German (de)	6568	602	1826
Greek (el)	17733	909	2805
Basque (eu)	4440	1418	5300
French (fr)	14377	1573	5011
Irish (ga)	257	322	1121
Hebrew (he)	14152	1254	3794
Hindi (hi)	282	289	1113
Italian (it)	10641	1202	3885
Polish (pl)	17731	1425	4391
Brazilian portuguese (pl)	23905	1976	6236
Romanian (ro)	10920	7714	38069
Swedish (sv)	1605	596	2103
Turkish (tr)	17945	1062	3304
Chinese (zh)	35326	1141	3462
Average	12563	1535	5887

Table 4.4: Number of training, development and testing examples in the PARSEME dataset for each language and an average for all languages.

4.2 Experimental setup

In the initial SemEval shared task, the experiments were conducted only in a multilingual setting *en+pt*, i.e., models were trained on the available training data, and then tested on the available testing data. For this thesis, we consider three additional novel settings for this task:

1. **en:** All three language models (mBERT, XLM-RoBERTa and mDeBERTa) are trained on only English training data;
2. **en_down:** All three language models are trained on only English downsampled training data;
3. **pt:** All three language models are trained on only Portuguese training data;
4. **en+pt:** All three language models are trained on all available training data; for both English and Portuguese.

In the first setting, we only train on English data. Since we only train on English data, when we test on Portuguese, or when we test on Galician data, this becomes a cross-lingual task where the models are evaluated on expressions in a language that was unseen during training. This helps us in answering our first research question by creating a cross-lingual setup to evaluate whether models are able to learn information about idiomaticity that is not language-specific. In the second setting, we only train on English data but that has been downsampled, i.e., the number of training instances in this setting is the same as the Portuguese training data. We do this to investigate whether the frequency of training data has a higher impact on model performance compared to the language, i.e., when we reduce the the number of training instances in English to the number of training instances in Portuguese, we check whether the model produces similar results as trained only on Portuguese data. Similar to the first setting, in the third setting, we only train on Portuguese data. Hence, when we test on English or on Galician data, this becomes a cross-lingual task. Similar to the first setting, it helps us in answering our first research question. Lastly, in the fourth setting, when we train on all available training data and test on Galician data, this becomes a cross-lingual task.

In the initial PARSEME shared task, experiments were conducted in a monolingual setting, i.e., models were trained on the training set for a particular language, and then tested on the same language. For this thesis, we consider further multilingual and cross-lingual settings.

1. **Mono:** We train a multilingual model on one language and test on the same language. We do this separately for each language.
2. **All:** We train a single multilingual model on the concatenation of the training data for all languages, and then test on each language. This helps us in answering our second research question by creating a setup that leverages training data from other languages and evaluates whether our models can use

this additional training data to improve performance of idiomaticity detection.

3. **Heldout:** For each language, a multilingual model is trained on training data from all other languages, and is then tested on that language that was held out during training. We again do this separately for each of the fourteen languages. This helps us in answering our first research question by creating a cross-lingual setup to evaluate whether models are able to learn information about idiomaticity that is not language-specific.

4.3 Implementation and parameter settings

We use Huggingface [110] implementations of mBERT, XLM-RoBERTa and mDeBERTa. Specifically, we use the *bert-base-multilingual-cased* implementation for mBERT, *xlm-roberta-base* implementation for XLM-RoBERTa and *mdeberta-v3-base* implementation for mDeBERTa. mBERT is pre-trained on the 104 languages with the largest Wikipedias. It contains 12 hidden layers and has a hidden size of 768. Each attention layer of mBERT has 12 attention heads while the model has 110 million parameters overall. XLM-RoBERTa and mDeBERTa are pre-trained on 2.5TB of CommonCrawl data covering 100 languages. XLM-RoBERTa is trained on the same architecture as BERT-base but has 270 million parameters [21] and also has a larger vocabulary size. mDeBERTa contains 12 hidden layers and has a hidden size of 768. It has 80 million backbone parameters with a vocabulary size of 250k and also has 190M million parameters in the embedding layer [37]. We use mBERT, mDeBERTa and XLM-RoBERTa for the SemEval task and mBERT for the PARSEME task. MTLB-STRUCT [100] system, which we use in the PARSEME task, was implemented using mBERT. Due to time constraints of training the system multiple times in the **mono** and **heldout** settings, we decided to restrict ourselves to using only mBERT in the PARSEME shared task.

For the SemEval task, since the gold standard for the test data was not publicly available when we conducted our experiments, we upload our models’ predictions to the competition website to obtain results over the test data.

For the MTLB-STRUCT system for the PARSEME task, we use the “multi-task” setting, where the loss of the model is back-propagated based on learning of MWE and dependency parse tags [101]. For both the multilingual and cross-lingual settings (described in Section 4.2), we use the default parameter settings of MTLB-STRUCT, where the number of epochs is 10 and the batch size is 3×10^{-5} .

While we use existing systems to answer our research questions, we implement novel cross-lingual and multilingual setups for each of the experiments to answer our research questions. We also implement a baseline for each task to compare it with our cross-lingual setup to see if models are able to learn information about idiomaticity that is not language specific.

4.4 Evaluation metrics

For the SemEval task, the classes are imbalanced. In addition to there being more English examples than Portuguese, there are also more idiomatic instances compared to literal instances in the training data as described in Section 4.1.1. We therefore follow the shared task and evaluate using macro-average F1 score which is the macro averaged harmonic mean of the precision and recall for each class.

For the PARSEME task, we also use the shared task evaluation metrics: global token-based F1 score, global MWE-based F1 score, and unseen MWE-based F1 score. The global token-based evaluation measures the precision and recall of the predicted VMWE boundaries. The global MWE-based evaluation measures the precision and recall of complete VMWEs, including their type (e.g., LVC, VPC). The unseen MWE-based evaluation considers only VMWEs that are not observed in the

training (or development) data. Note that in the case of cross-lingual experiments in the heldout setting, in which systems are evaluated on expressions in a language that was not observed during training, all test expressions are unseen during training. For both tasks we compare against a most-frequent class baseline. For the PARSEME task, for each language, we label each token as the most-frequent class of VMWE observed in the training data for that language.

Chapter 5

Results

Here we present results on the SemEval (Section 5.1) and then PARSEME (Section 5.2) tasks.

5.1 SemEval

Results are shown in Table 5.1 for the SemEval task, where the results are rounded to 3 decimal places. We focus on cross-lingual settings, i.e., when the model is tested on a different language than it is trained on.

When testing on English, and training on Portuguese, each model improves over the most-frequent class baseline, although the difference is quite small for mBERT. When testing on Portuguese, and training on English, the findings are similar in that all models again improve over the baseline. It is also interesting to note that for mBERT and RoBERTa, results for training on English and testing on Portuguese are in fact higher than for training and testing on Portuguese. This could be due to the larger number of training instances for English compared to Portuguese (section 4.1). When testing on Galician, results for models trained on English do not improve over the baseline. Models trained on Portuguese perform better than those trained on English, and show small improvements over the baseline. Despite differences in training data

Model	Train	Test			
		en	pt	gl	ALL
mBERT	en	0.717	0.583	0.420	0.587
	en_down	0.680	0.570	0.467	0.584
	pt	0.355	0.578	0.478	0.482
	en+pt	0.700	0.662	0.550	0.665
RoBERTa	en	0.697	0.590	0.390	0.571
	en_down	0.658	0.5435	0.387	0.541
	pt	0.555	0.553	0.440	0.531
	en+pt	0.706	0.668	0.526	0.651
mDeBERTa	en	0.700	0.523	0.304	0.526
	en_down	0.700	0.548	0.427	0.536
	pt	0.582	0.567	0.499	0.556
	en+pt	0.720	0.644	0.495	0.635
Baseline		0.345	0.391	0.434	0.389

Table 5.1: Macro-average F1 score for each model, training and testing on the indicated language(s). Results for a most-frequent class baseline are also shown.

size for English and Portuguese, models trained on Portuguese could perform better on Galician than those trained on English because Portuguese and Galician are both Romance languages. Training on the concatenation of the English and Portuguese training data gives the best results on Galician, and improves over the results for models trained on only Portuguese for mBERT and RoBERTa. This finding suggests that models for predicting idiomaticity can be improved with additional training data from other languages.

To understand if there is any influence of training size in the cross-lingual experiments, we performed downsampled experiments on the English training data. As noted in Table 4.2, there are more training instances in English compared to Portuguese. Hence, we reduced the training size of the English only data to the size of the Portuguese size and trained our models again, resulting in 1164 training instances compared to 3327 as in the original *en* setting. These training instances in the undersampled dataset were sampled randomly. Results of these experiments are shown in Table 5.1, where under each model, the setting *en_down* contains the results. For mBERT, overall performance dropped as expected but not substantially. Similarly,

for RoBERTa, there is a drop in performance when the data is downsampled. Interestingly, for mDeBERTa, there is an overall increase in performance, particularly when tested on Galician data. This shows that the frequency of training data among languages does not have a substantial impact on model performance. This differs from our previous result about adding training data to improve model performance in predicting idiomaticity as the previous result discusses about adding data from different languages while this observation relates to having more training instances in the same language. This indicates that diversity among languages in the training data improves model performance compared to the volume of training data.

Overall, these findings indicate that the models are able to learn information about idiomaticity that is not language-specific, answering our first research question. We show that our models learn information about idiomaticity when trained on one or more languages and use that information when tested on instances from a different language. This can be particularly useful in cases involving low-resource languages. In low-resource languages where there is little training data compared to high-resource languages, we can apply our novel cross-lingual setup to learn information about idiomaticity from high-resource languages such as English or Portuguese and test them on expressions from low-resource languages such as Galician. These findings are in line with those of [27].

5.2 PARSEME

Results on the PARSEME task are shown in Table 5.2, where the results are again rounded to 3 decimal places. The monolingual approach (“Mono” in 5.2) is our reproduction of the MTLB-STRUCT system on the shared task. In this setting, a monolingual model is trained and tested on each language. In the “all” setting, a model is trained on the concatenation of the training data for all languages. For

Language	Setting	MWE	Token	Unseen
DE	Mono	0.699	0.734	0.398
	All	0.729	0.738	0.434
	Heldout	0.269	0.423	0.207
EL	Mono	0.732	0.776	0.420
	All	0.743	0.776	0.423
	Heldout	0.407	0.415	0.147
EU	Mono	0.804	0.832	0.346
	All	0.815	0.839	0.380
	Heldout	0.194	0.258	0.112
FR	Mono	0.802	0.830	0.431
	All	0.797	0.825	0.437
	Heldout	0.501	0.560	0.196
GA	Mono	0.311	0.465	0.210
	All	0.422	0.483	0.301
	Heldout	0.111	0.133	0.069
HE	Mono	0.482	0.527	0.215
	All	0.491	0.536	0.219
	Heldout	0.141	0.146	0.064
HI	Mono	0.729	0.785	0.504
	All	0.759	0.796	0.549
	Heldout	0.376	0.452	0.278
IT	Mono	0.632	0.673	0.227
	All	0.618	0.656	0.200
	Heldout	0.376	0.437	0.160
PL	Mono	0.815	0.826	0.400
	All	0.808	0.815	0.380
	Heldout	0.361	0.382	0.144
PT	Mono	0.736	0.758	0.358
	All	0.807	0.821	0.397
	Heldout	0.486	0.500	0.183
RO	Mono	0.903	0.908	0.299
	All	0.898	0.900	0.275
	Heldout	0.481	0.502	0.092
SV	Mono	0.721	0.731	0.425
	All	0.769	0.751	0.467
	Heldout	0.303	0.413	0.215
TR	Mono	0.701 [*]	0.716	0.430
	All	0.708	0.718	0.457
	Heldout	0.394	0.416	0.189
ZH	Mono	0.696	0.725	0.605
	All	0.705	0.732	0.618
	Heldout	0.121	0.188	0.148
Average	Mono	0.699	0.738	0.380
	All	0.722	0.746	0.400
	Heldout	0.331	0.381	0.169
	Baseline	0.002	0.067	0.001

Table 5.2: MWE-based, token-based, and unseen F1 score for the monolingual (mono), “all”, and “heldout”, experimental settings, for each language. The “Average” across languages is also calculated.

Category	Frequency	Mono	All	Heldout
IRV	11571	0.6945	0.7188	0.3135
LVC.cause	1809	0.3965	0.4429	0.0994
LVC.full	24574	0.6392	0.6661	0.3495
VID	18553	0.5147	0.5335	0.2320
VPC.full	3018	0.5799	0.5825	0.0565
VPC.semi	4204	0.4363	0.4712	0.0052
MVC	4057	0.4707	0.4853	0.0000
IAV	680	0.4929	0.5408	0.0000
LS.ICV	37	0.0000	0.0000	0.0000

Table 5.3: Frequency of each category and per-category MWE-based F1 score across languages which have instances of these categories.

“heldout”, for a given target language, a model is trained on all other languages, and then evaluated on the target language, which was held out during training. When calculating the unseen MWE-based F1 score (“Unseen” in Table 5.2), for each setting, we report results over the instances that are unseen based on the monolingual training and development data. This enables comparisons between settings for this evaluation metric. However, in the heldout setting, all test instances are in fact unseen during training.

For each of the three evaluation metrics, we see that the average F1 score for the “all” setting is higher than that for the monolingual setting. This indicates that information from other languages can be leveraged to give improvements over a monolingual approach, answering our second research question. This is inline with the findings on the SemEval task from Section 5.1. We also see that, for all languages, and all evaluation metrics, the F1 score for the heldout setting is less than that for the monolingual setting. This is perhaps unsurprising; a model that has access to language-specific training data is able to outperform one that does not. However, the results in the heldout setting are higher than the baseline on average as shown in Table 5.2. This indicates that models are able to learn information about MWEs that is not language specific. Again, this can be particularly useful in cases involving

low-resource languages. This is again inline with the findings on the SemEval task from Section 5.1 and the findings of [27].

To better understand the performance in the heldout setting, we report results for each category of VMWE in Table 5.3, where the results are rounded to four decimal places. The best results for the heldout setting are for (full) light-verb constructions (LVC.full), inherently-reflexive verbs (IRV), and verbal idioms (VID). Although not all languages have instances of all of these categories, they are by far the most frequent categories of VMWEs in the PARSEME 1.2 data [78], which could be why the model performs relatively well on these categories in the heldout setting.

Chapter 6

Conclusions

Multiword expressions are difficult to identify due to them exhibiting idiomaticity. Knowledge of MWEs is important for downstream NLP tasks such as machine translation and sentiment analysis. While a lot of work has focused on monolingual settings, little work has been on cross-lingual “zero-shot” settings. In this thesis, we considered new cross-lingual settings for the SemEval 2022 task 2 subtask A and PARSEME 1.2 shared tasks, in which models are evaluated on languages that are not seen during training. We also explored if adding additional training data from other languages improves model performance. The first research question is *Can models for automatically predicting idiomaticity generalize to MWEs in a different language that was not seen during training?* Our findings indicate that language models are able to learn information about MWEs and idiomaticity that is not language-specific. The *heldout* setting in Table 5.2 shows that the models outperform our baseline in a cross-lingual setup thereby showing that models are able to learn information about idiomaticity in one language and apply it to other languages. This can also be seen in Table 5.1 where in multiple settings; the models outperform the baseline implemented. This can be particularly useful when low-resource languages are involved. We can use training data from high-resource languages to learn information

about idiomaticity and use it to identify and classify expressions in low-resource languages. The second research question is *Can data from other languages be leveraged to improve performance of idiomaticity detection?*. We considered new multilingual settings for the PARSEME 1.2 shared task where data from other languages was combined. Our findings show that additional training data from other languages can be leveraged to give improvements over monolingual models for identifying MWEs and predicting idiomaticity. This is highlighted in the *all* setting in Table 5.2, where the average across each of the settings is the highest for the *all* setting over others and, in particular, is higher than the monolingual setting. This is also highlighted in the SemEval shared task where the *en+pt* setting in Table 5.1 has the best results compared to the other settings, which only train on one of English or Portuguese, for all the models. This can be useful for improving the performance of a monolingual system by incorporating additional training data from other languages.

The contributions of this thesis are highlighted as follows:

1. Proposed novel cross-lingual setups for the PARSEME and SemEval tasks.
2. Showed that models in a cross-lingual setting outperform baselines, demonstrating that models for predicting idiomaticity can generalize to MWEs in languages that were unseen during training.
3. Demonstrated that models can leverage additional training data from other languages to improve model performance by concatenating the training data in the other languages. This improves over initial monolingual settings.

For future work, we can explore several different avenues. We intend to consider more powerful multilingual language models for PARSEME 1.2 (e.g., XLM-RoBERTa, mDeBERTa). Also, we use only the base versions of mBERT and XLM-RoBERTa. Instead, we can use larger versions of mBERT and XLM-RoBERTa as shown in [16] that could improve the performance of our systems. There is also scope for hyper-

parameter optimization such as in number of epochs, the batch size and the learning rate for the PARSEME task.

We also intend to test our cross-lingual setting on the very recent PARSEME 1.3 shared task [86]. This edition contains multiple changes over PARSEME 1.2. It introduces new languages, Arabic and Serbian. It also enlarges the Chinese, Greek and Swedish datasets by adding more instances. Certain languages such as Croatian and Romanian also have improvements by adding sentences that contained types of MWE were not previously included, such as inherently adpositional verbs (IAVs). PARSEME 1.3 also provides new annotations for certain languages to improve quality.

Another interesting line of work we can pursue is to see how much context affects the model’s performance in a cross-lingual setting. [16] showed that while fine-tuning their models in the SemEval shared task, not adding the context, i.e., not including the previous and the next sentences, led to an improvement in the model performance. We could also remove the context while fine-tuning the model thereby decreasing the training time and potentially improving the performance on classification of unseen VMWEs.

As of recently, there has been an increase in large language models with billions of parameters such as GPT-3 [13] and BLOOMZ [89]. These models have shown to have state-of-the-art performance in cross-lingual tasks such as machine translation [116]. We would like to evaluate these models to see if they can also capture information about idiomaticity in a cross-lingual setting.

Another line of work we can pursue is the use of prompt-based learning [55] for idiomaticity prediction and identifying MWEs, where given an input x , the input is transformed into a sentence x' using an existing template or ‘prompt’ with empty slots. The model predicts the word with the highest probability in the empty slots and compares it with the actual word that needs to be filled in. This can be especially

useful when there is no supervised training data.

Bibliography

- [1] Otavio Acosta, Aline Villavicencio, and Viviane Moreira, *Identification and treatment of multiword expressions applied to information retrieval*, Proceedings of the Workshop on Multiword Expressions: from Parsing and Generation to the Real World (Portland, Oregon, USA), Association for Computational Linguistics, June 2011, pp. 101–109.
- [2] Felipe Almeida and Geraldo Xexéo, *Word embeddings: A survey*, 2023.
- [3] Wissam Antoun, Fady Baly, and Hazem Hajj, *AraBERT: Transformer-based model for Arabic language understanding*, Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection (Marseille, France), European Language Resource Association, May 2020, pp. 9–15 (English).
- [4] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio, *Neural machine translation by jointly learning to align and translate*, arXiv preprint arXiv:1409.0473 (2014).
- [5] Timothy Baldwin, Colin Bannard, Takaaki Tanaka, and Dominic Widdows, *An empirical model of multiword expression decomposability*, Proceedings of the ACL 2003 Workshop on Multiword Expressions: Analysis, Acquisition and Treatment (Sapporo, Japan), Association for Computational Linguistics, July 2003, pp. 89–96.

- [6] Timothy Baldwin and Su Nam Kim, *Multiword expressions*, Handbook of Natural Language Processing (Nitin Indurkha and Fred J. Damerau, eds.), CRC Press, Boca Raton, USA, 2nd ed., 2010.
- [7] Marco Baroni, Georgiana Dinu, and Germán Kruszewski, *Don't count, predict! a systematic comparison of context-counting vs. context-predicting semantic vectors*, Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (Baltimore, Maryland), Association for Computational Linguistics, June 2014, pp. 238–247.
- [8] Nuria Bel, Cornelis HA Koster, and Marta Villegas, *Cross-lingual text categorization*, Research and Advanced Technology for Digital Libraries: 7th European Conference, ECDL 2003 Trondheim, Norway, August 17-22, 2003 Proceedings 7, Springer, 2003, pp. 126–139.
- [9] Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin, *A neural probabilistic language model*, J. Mach. Learn. Res. **3** (2003), no. null, 1137–1155.
- [10] Gábor Berend, *Opinion expression mining by exploiting keyphrase extraction*, Proceedings of 5th International Joint Conference on Natural Language Processing (Chiang Mai, Thailand), Asian Federation of Natural Language Processing, November 2011, pp. 1162–1170.
- [11] Archana Bhatia, Choh Man Teng, and James Allen, *Compositionality in verb-particle constructions*, Proceedings of the 13th Workshop on Multiword Expressions (MWE 2017) (Valencia, Spain), Association for Computational Linguistics, April 2017, pp. 139–148.

- [12] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov, *Enriching word vectors with subword information*, Transactions of the Association for Computational Linguistics **5** (2017), 135–146.
- [13] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al., *Language models are few-shot learners*, Advances in neural information processing systems **33** (2020), 1877–1901.
- [14] Marine Carpuat and Mona Diab, *Task-based evaluation of multiword expressions: a pilot study in statistical machine translation*, Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics (Los Angeles, California), Association for Computational Linguistics, June 2010, pp. 242–245.
- [15] Kyunghyun Cho, Bart van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio, *On the properties of neural machine translation: Encoder–decoder approaches*, Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation (Doha, Qatar), Association for Computational Linguistics, October 2014, pp. 103–111.
- [16] Zheng Chu, Ziqing Yang, Yiming Cui, Zhigang Chen, and Ming Liu, *HIT at SemEval-2022 task 2: Pre-trained language model for idioms detection*, Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022) (Seattle, United States), Association for Computational Linguistics, July 2022, pp. 221–227.
- [17] Junyoung Chung, Kyunghyun Cho, and Yoshua Bengio, *A character-level decoder without explicit segmentation for neural machine translation*, Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics

- (Volume 1: Long Papers) (Berlin, Germany), Association for Computational Linguistics, August 2016, pp. 1693–1703.
- [18] Kenneth Ward Church and Patrick Hanks, *Word association norms, mutual information, and lexicography*, Computational Linguistics **16** (1990), no. 1, 22–29.
- [19] Kevin Clark, Minh-Thang Luong, Quoc V Le, and Christopher D Manning, *Electra: Pre-training text encoders as discriminators rather than generators*, arXiv preprint arXiv:2003.10555 (2020).
- [20] Ronan Collobert and Jason Weston, *A unified architecture for natural language processing: Deep neural networks with multitask learning*, Proceedings of the 25th international conference on Machine learning, 2008, pp. 160–167.
- [21] Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov, *Unsupervised cross-lingual representation learning at scale*, Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (Online), Association for Computational Linguistics, July 2020, pp. 8440–8451.
- [22] Alexis Conneau, Ruty Rinott, Guillaume Lample, Adina Williams, Samuel Bowman, Holger Schwenk, and Veselin Stoyanov, *XNLI: Evaluating cross-lingual sentence representations*, Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (Brussels, Belgium), Association for Computational Linguistics, October–November 2018, pp. 2475–2485.
- [23] Silvio Cordeiro, Aline Villavicencio, Marco Idiart, and Carlos Ramisch, *Unsupervised compositionality prediction of nominal compounds*, Computational Linguistics **45** (2019), no. 1, 1–57.

- [24] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova, *BERT: Pre-training of deep bidirectional transformers for language understanding*, Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers) (Minneapolis, Minnesota), Association for Computational Linguistics, June 2019, pp. 4171–4186.
- [25] Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon, *Unified language model pre-training for natural language understanding and generation*, Advances in neural information processing systems **32** (2019).
- [26] Majid F. Sadi, *Word sense disambiguation using vector space models*, Ph.D. thesis, University of Zanjan, 04 2019.
- [27] Samin Fakharian and Paul Cook, *Contextualized embeddings encode monolingual and cross-lingual knowledge of idiomaticity*, Proceedings of the 17th Workshop on Multiword Expressions (MWE 2021) (Online), Association for Computational Linguistics, August 2021, pp. 23–32.
- [28] Afsaneh Fazly, Paul Cook, and Suzanne Stevenson, *Unsupervised type and token identification of idiomatic expressions*, Computational Linguistics **35** (2009), no. 1, 61–103.
- [29] Lev Finkelstein, Evgeniy Gabrilovich, Yossi Matias, Ehud Rivlin, Zach Solan, Gadi Wolfman, and Eytan Ruppin, *Placing search in context: The concept revisited*, Proceedings of the 10th international conference on World Wide Web, 2001, pp. 406–414.
- [30] Marcos Garcia, Tiago Kramer Vieira, Carolina Scarton, Marco Idiart, and Aline Villavicencio, *Assessing the representations of idiomaticity in vector*

models with a noun compound dataset labeled at type and token levels, Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers) (Online), Association for Computational Linguistics, August 2021, pp. 2730–2741.

- [31] Waseem Gharbieh, Virendrakumar Bhavsar, and Paul Cook, *Deep learning models for multiword expression identification*, Proceedings of the 6th Joint Conference on Lexical and Computational Semantics (*SEM 2017) (Vancouver, Canada), Association for Computational Linguistics, August 2017, pp. 54–64.
- [32] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio, *Generative adversarial networks*, 2014.
- [33] Alex Graves, *Generating sequences with recurrent neural networks*, arXiv preprint arXiv:1308.0850 (2013).
- [34] Hessel Haagsma, Malvina Nissim, and Johan Bos, *The other side of the coin: Unsupervised disambiguation of potentially idiomatic expressions by contrasting senses*, Proceedings of the Joint Workshop on Linguistic Annotation, Multiword Expressions and Constructions (LAW-MWE-CxG-2018) (Santa Fe, New Mexico, USA), Association for Computational Linguistics, August 2018, pp. 178–184.
- [35] Ali Hakimi Parizi and Paul Cook, *Joint training for learning cross-lingual embeddings with sub-word information without parallel corpora*, Proceedings of the Ninth Joint Conference on Lexical and Computational Semantics (Barcelona, Spain (Online)), Association for Computational Linguistics, December 2020, pp. 39–49.

- [36] Reyhaneh Hashempour and Aline Villavicencio, *Leveraging contextual embeddings and idiom principle for detecting idiomaticity in potentially idiomatic expressions*, Proceedings of the Workshop on the Cognitive Aspects of the Lexicon (Online), Association for Computational Linguistics, December 2020, pp. 72–80.
- [37] Pengcheng He, Jianfeng Gao, and Weizhu Chen, *Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing*, arXiv preprint arXiv:2111.09543 (2021).
- [38] Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen, *Deberta: Decoding-enhanced bert with disentangled attention*, arXiv preprint arXiv:2006.03654 (2020).
- [39] Sepp Hochreiter, *The vanishing gradient problem during learning recurrent neural nets and problem solutions*, International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems **6** (1998), no. 02, 107–116.
- [40] Sepp Hochreiter and Jürgen Schmidhuber, *Long short-term memory*, Neural computation **9** (1997), no. 8, 1735–1780.
- [41] Zhuolin Jiang, Amro El-Jaroudi, William Hartmann, Damianos Karakos, and Lingjun Zhao, *Cross-lingual information retrieval with BERT*, Proceedings of the workshop on Cross-Language Search and Summarization of Text and Speech (CLSSTS2020) (Marseille, France), European Language Resources Association, May 2020, pp. 26–31 (English).
- [42] Nal Kalchbrenner and Phil Blunsom, *Recurrent continuous translation models*, Proceedings of the 2013 conference on empirical methods in natural language processing, 2013, pp. 1700–1709.

- [43] Milton King and Paul Cook, *Leveraging distributed representations and lexicosyntactic fixedness for token-level prediction of the idiomaticity of English verb-noun combinations*, Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers) (Melbourne, Australia), Association for Computational Linguistics, July 2018, pp. 345–350.
- [44] Diederik P Kingma and Jimmy Ba, *Adam: A method for stochastic optimization*, arXiv preprint arXiv:1412.6980 (2014).
- [45] Angelika Kirilin, Felix Krauss, and Yannick Versley, *ICL-HD at SemEval-2016 task 10: Improving the detection of minimal semantic units and their meanings with an ontology and word embeddings*, Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016) (San Diego, California), Association for Computational Linguistics, June 2016, pp. 937–945.
- [46] Ioannis Korkontzelos and Suresh Manandhar, *Detecting compositionality in multi-word expressions.*, Proceedings of the ACL-IJCNLP 2009 Conference Short Papers (Suntec, Singapore), Association for Computational Linguistics, August 2009, pp. 65–68.
- [47] Yuri Kuratov and Mikhail Arkhipov, *Adaptation of deep bidirectional multilingual transformers for russian language*, arXiv preprint arXiv:1905.07213 (2019).
- [48] Murathan Kurfah and Robert Östling, *Disambiguation of potentially idiomatic expressions with contextual embeddings*, Proceedings of the Joint Workshop on Multiword Expressions and Electronic Lexicons (online), Association for Computational Linguistics, December 2020, pp. 85–94.
- [49] Guillaume Lample and Alexis Conneau, *Cross-lingual language model pretraining*, arXiv preprint arXiv:1901.07291 (2019).

- [50] Eric Laporte, Takuya Nakamura, and Stavroula Voyatzi, *A french corpus annotated for multiword expressions with adverbial function*, Language Resources and Evaluation Conference (LREC). Linguistic Annotation Workshop, 2008, pp. 48–51.
- [51] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer, *BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension*, Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (Online), Association for Computational Linguistics, July 2020, pp. 7871–7880.
- [52] Patrick Lewis, Barlas Oguz, Ruty Rinott, Sebastian Riedel, and Holger Schwenk, *MLQA: Evaluating cross-lingual extractive question answering*, Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (Online), Association for Computational Linguistics, July 2020, pp. 7315–7330.
- [53] Wang Ling, Isabel Trancoso, Chris Dyer, and Alan W Black, *Character-based neural machine translation*, arXiv preprint arXiv:1511.04586 (2015).
- [54] Changsheng Liu and Rebecca Hwa, *Heuristically informed unsupervised idiom usage recognition*, Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (Brussels, Belgium), Association for Computational Linguistics, October-November 2018, pp. 1723–1731.
- [55] Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig, *Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing*, ACM Computing Surveys **55** (2023), no. 9, 1–35.

- [56] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov, *Roberta: A robustly optimized bert pretraining approach*, arXiv preprint arXiv:1907.11692 (2019).
- [57] Thang Luong, Richard Socher, and Christopher Manning, *Better word representations with recursive neural networks for morphology*, Proceedings of the Seventeenth Conference on Computational Natural Language Learning (Sofia, Bulgaria), Association for Computational Linguistics, August 2013, pp. 104–113.
- [58] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean, *Efficient estimation of word representations in vector space*, arXiv preprint arXiv:1301.3781 (2013).
- [59] Tomas Mikolov, Martin Karafiát, Lukas Burget, Jan Cernocký, and Sanjeev Khudanpur, *Recurrent neural network based language model.*, Interspeech, vol. 2, Makuhari, 2010, pp. 1045–1048.
- [60] Tomas Mikolov, Quoc V Le, and Ilya Sutskever, *Exploiting similarities among languages for machine translation*, arXiv preprint arXiv:1309.4168 (2013).
- [61] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean, *Distributed representations of words and phrases and their compositionality*, Advances in neural information processing systems **26** (2013).
- [62] Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig, *Linguistic regularities in continuous space word representations*, Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Atlanta, Georgia), Association for Computational Linguistics, June 2013, pp. 746–751.

- [63] Andriy Mnih and Geoffrey E Hinton, *A scalable hierarchical distributed language model*, Advances in neural information processing systems **21** (2008).
- [64] Frederic Morin and Yoshua Bengio, *Hierarchical probabilistic neural network language model*, International workshop on artificial intelligence and statistics, PMLR, 2005, pp. 246–252.
- [65] Rohit Mundra, Emma Peng, Richard Socher, and Ajay Sohmshetty, *CS224n: Natural Language Processing with Deep Learning 1 Lecture Notes: Part II*, https://web.stanford.edu/class/archive/cs/cs224n/cs224n.1174/lecture_notes/cs224n-2017-notes2.pdf, 2017, Accessed on 2023-06-12.
- [66] Marwa Naili, Anja Habacha Chaibi, and Henda Hajjami Ben Ghezala, *Comparative study of word embedding methods in topic segmentation*, Procedia computer science **112** (2017), 340–349.
- [67] Navnita Nandakumar, Timothy Baldwin, and Bahar Salehi, *How well do embedding models capture non-compositionality? a view from multiword expressions*, Proceedings of the 3rd Workshop on Evaluating Vector Space Representations for NLP (Minneapolis, USA), Association for Computational Linguistics, June 2019, pp. 27–34.
- [68] Niki Parmar, Ashish Vaswani, Jakob Uszkoreit, Lukasz Kaiser, Noam Shazeer, Alexander Ku, and Dustin Tran, *Image transformer*, International conference on machine learning, PMLR, 2018, pp. 4055–4064.
- [69] Jeffrey Pennington, Richard Socher, and Christopher Manning, *GloVe: Global vectors for word representation*, Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP) (Doha, Qatar), Association for Computational Linguistics, October 2014, pp. 1532–1543.

- [70] Matthew E. Peters, Waleed Ammar, Chandra Bhagavatula, and Russell Power, *Semi-supervised sequence tagging with bidirectional language models*, Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (Vancouver, Canada), Association for Computational Linguistics, July 2017, pp. 1756–1765.
- [71] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer, *Deep contextualized word representations*, Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers) (New Orleans, Louisiana), Association for Computational Linguistics, June 2018, pp. 2227–2237.
- [72] Thomas Pickard, *Comparing word2vec and GloVe for automatic measurement of MWE compositionality*, Proceedings of the Joint Workshop on Multiword Expressions and Electronic Lexicons (online), Association for Computational Linguistics, December 2020, pp. 95–100.
- [73] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al., *Improving language understanding by generative pre-training*, (2018).
- [74] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al., *Language models are unsupervised multitask learners*, OpenAI blog **1** (2019), no. 8, 9.
- [75] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu, *Exploring the limits of transfer learning with a unified text-to-text transformer*, The Journal of Machine Learning Research **21** (2020), no. 1, 5485–5551.

- [76] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang, *SQuAD: 100,000+ questions for machine comprehension of text*, Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (Austin, Texas), Association for Computational Linguistics, November 2016, pp. 2383–2392.
- [77] Carlos Ramisch, Silvio Ricardo Cordeiro, Agata Savary, Veronika Vincze, Verginica Barbu Mititelu, Archana Bhatia, Maja Buljan, Marie Candito, Polona Gantar, Voula Giouli, Tunga Güngör, Abdelati Hawwari, Uxoá Iñurrieta, Jolanta Kovalevskaitė, Simon Krek, Timm Lichte, Chaya Liebeskind, Johanna Monti, Carla Parra Escartín, Behrang QasemiZadeh, Renata Ramisch, Nathan Schneider, Ivelina Stoyanova, Ashwini Vaidya, and Abigail Walsh, *Edition 1.1 of the PARSEME shared task on automatic identification of verbal multiword expressions*, Proceedings of the Joint Workshop on Linguistic Annotation, Multiword Expressions and Constructions (LAW-MWE-CxG-2018) (Santa Fe, New Mexico, USA), Association for Computational Linguistics, August 2018, pp. 222–240.
- [78] Carlos Ramisch, Agata Savary, Bruno Guillaume, Jakub Waszczuk, Marie Candito, Ashwini Vaidya, Verginica Barbu Mititelu, Archana Bhatia, Uxoá Iñurrieta, Voula Giouli, Tunga Güngör, Menghan Jiang, Timm Lichte, Chaya Liebeskind, Johanna Monti, Renata Ramisch, Sara Stymne, Abigail Walsh, and Hongzhi Xu, *Edition 1.2 of the PARSEME shared task on semi-supervised identification of verbal multiword expressions*, Proceedings of the Joint Workshop on Multiword Expressions and Electronic Lexicons (online), Association for Computational Linguistics, December 2020, pp. 107–118.
- [79] Siva Reddy, Diana McCarthy, and Suresh Manandhar, *An empirical study on compositionality in compound nouns*, Proceedings of 5th International Joint

- Conference on Natural Language Processing (Chiang Mai, Thailand), Asian Federation of Natural Language Processing, November 2011, pp. 210–218.
- [80] Zhixiang Ren, Yajuan Lü, Jie Cao, Qun Liu, and Yun Huang, *Improving statistical machine translation using domain bilingual multiword expressions*, Proceedings of the Workshop on Multiword Expressions: Identification, Interpretation, Disambiguation and Applications (MWE 2009) (Singapore), Association for Computational Linguistics, August 2009, pp. 47–54.
- [81] Sebastian Ruder, Ivan Vulić, and Anders Søgaard, *A survey of cross-lingual word embedding models*, Journal of Artificial Intelligence Research **65** (2019), 569–631.
- [82] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams, *Learning representations by back-propagating errors*, nature **323** (1986), no. 6088, 533–536.
- [83] Alexander Rush, *Torch-struct: Deep structured prediction library*, Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations (Online), Association for Computational Linguistics, July 2020, pp. 335–342.
- [84] Ivan A Sag, Timothy Baldwin, Francis Bond, Ann Copestake, and Dan Flickinger, *Multiword expressions: A pain in the neck for nlp*, Computational Linguistics and Intelligent Text Processing: Third International Conference, CICLing 2002 Mexico City, Mexico, February 17–23, 2002 Proceedings 3, Springer, 2002, pp. 1–15.
- [85] Giancarlo Salton, Robert Ross, and John Kelleher, *Idiom token classification using sentential distributed semantics*, Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)

- (Berlin, Germany), Association for Computational Linguistics, August 2016, pp. 194–204.
- [86] Agata Savary, Cherifa Ben Khelil, Carlos Ramisch, Voula Giouli, Verginica Barbu Mititelu, Najet Hadj Mohamed, Cvetana Krstev, Chaya Liebeskind, Hongzhi Xu, Sara Stymne, Tunga Güngör, Thomas Pickard, Bruno Guillaume, Eduard Bejček, Archana Bhatia, Marie Candito, Polona Gantar, Uxo Inurrieta, Albert Gatt, Jolanta Kovalevskaite, Timm Lichte, Nikola Ljubešić, Johanna Monti, Carla Parra Escartín, Mehrnoush Shamsfard, Ivelina Stoyanova, Veronika Vincze, and Abigail Walsh, *PARSEME corpus release 1.3*, Proceedings of the 19th Workshop on Multiword Expressions (MWE 2023) (Dubrovnik, Croatia), Association for Computational Linguistics, May 2023, pp. 24–35.
- [87] Agata Savary, Silvio Cordeiro, and Carlos Ramisch, *Without lexicons, multiword expression identification will never fly: A position statement*, Proceedings of the Joint Workshop on Multiword Expressions and WordNet (MWE-WN 2019) (Florence, Italy), Association for Computational Linguistics, August 2019, pp. 79–91.
- [88] Agata Savary, Carlos Ramisch, Silvio Cordeiro, Federico Sangati, Veronika Vincze, Behrang QasemiZadeh, Marie Candito, Fabienne Cap, Voula Giouli, Ivelina Stoyanova, and Antoine Doucet, *The PARSEME shared task on automatic identification of verbal multiword expressions*, Proceedings of the 13th Workshop on Multiword Expressions (MWE 2017) (Valencia, Spain), Association for Computational Linguistics, April 2017, pp. 31–47.
- [89] Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon,

Matthias Gallé, et al., *Bloom: A 176b-parameter open-access multilingual language model*, arXiv preprint arXiv:2211.05100 (2022).

- [90] Andreas Scherbakov, Ekaterina Vylomova, Fei Liu, and Timothy Baldwin, *VectorWeavers at SemEval-2016 task 10: From incremental meaning to semantic unit (phrase by phrase)*, Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016) (San Diego, California), Association for Computational Linguistics, June 2016, pp. 946–952.
- [91] Dominik Schlechtweg, Anna HäTTY, Marco Del Tredici, and Sabine Schulte im Walde, *A wind of change: Detecting and evaluating lexical semantic change across times and domains*, Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (Florence, Italy), Association for Computational Linguistics, July 2019, pp. 732–746.
- [92] Nathan Schneider, Dirk Hovy, Anders Johannsen, and Marine Carpuat, *SemEval-2016 task 10: Detecting minimal semantic units and their meanings (DiMSUM)*, Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016) (San Diego, California), Association for Computational Linguistics, June 2016, pp. 546–559.
- [93] Nathan Schneider and Noah A. Smith, *A corpus and model integrating multi-word expressions and supersenses*, Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Denver, Colorado), Association for Computational Linguistics, May–June 2015, pp. 1537–1547.
- [94] Holger Schwenk and Xian Li, *A corpus for multilingual document classification in eight languages*, Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018) (Miyazaki, Japan), European Language Resources Association (ELRA), May 2018.

- [95] Rico Sennrich, Barry Haddow, and Alexandra Birch, *Neural machine translation of rare words with subword units*, Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (Berlin, Germany), Association for Computational Linguistics, August 2016, pp. 1715–1725.
- [96] Buck Shlegeris, Fabien Roger, Lawrence Chan, and Euan McLean, *Language models are better than humans at next-token prediction*, arXiv preprint arXiv:2212.11281 (2022).
- [97] Vered Shwartz and Ido Dagan, *Still a pain in the neck: Evaluating text representations on lexical composition*, Transactions of the Association for Computational Linguistics **7** (2019), 403–419.
- [98] Peter Sutor, Yiannis Aloimonos, Cornelia Fermuller, and Douglas Summers-Stay, *Metaconcepts: isolating context in word embeddings*, 2019 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR), IEEE, 2019, pp. 544–549.
- [99] Ilya Sutskever, Oriol Vinyals, and Quoc V Le, *Sequence to sequence learning with neural networks*, Advances in neural information processing systems **27** (2014).
- [100] Shiva Taslimipoor, Sara Bahaadini, and Ekaterina Kochmar, *MTLB-STRUCT @parseme 2020: Capturing unseen multiword expressions using multi-task learning and pre-trained masked language models*, Proceedings of the Joint Workshop on Multiword Expressions and Electronic Lexicons (online), Association for Computational Linguistics, December 2020, pp. 142–148.
- [101] Shiva Taslimipoor, Omid Rohanian, and Le An Ha, *Cross-lingual transfer learning and multitask learning for capturing multiword expressions*, Proceed-

- ings of the Joint Workshop on Multiword Expressions and WordNet (MWE-WN 2019) (Florence, Italy), Association for Computational Linguistics, August 2019, pp. 155–161.
- [102] Harish Tayyar Madabushi, Edward Gow-Smith, Marcos Garcia, Carolina Scarton, Marco Idiart, and Aline Villavicencio, *SemEval-2022 task 2: Multilingual idiomaticity detection and sentence embedding*, Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022) (Seattle, United States), Association for Computational Linguistics, July 2022, pp. 107–121.
- [103] Simone Tedeschi and Roberto Navigli, *NER4ID at SemEval-2022 task 2: Named entity recognition for idiomaticity detection*, Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022) (Seattle, United States), Association for Computational Linguistics, July 2022, pp. 204–210.
- [104] Ian Tenney, Dipanjan Das, and Ellie Pavlick, *BERT rediscovers the classical NLP pipeline*, Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (Florence, Italy), Association for Computational Linguistics, July 2019, pp. 4593–4601.
- [105] Alan M Turing, *Computing machinery and intelligence*, Springer, 2009.
- [106] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin, *Attention is all you need*, Advances in neural information processing systems **30** (2017).
- [107] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman, *GLUE: A multi-task benchmark and analysis platform for natural language understanding*, Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP (Brussels,

- Belgium), Association for Computational Linguistics, November 2018, pp. 353–355.
- [108] Joseph Weizenbaum, *Eliza—a computer program for the study of natural language communication between man and machine*, Communications of the ACM **9** (1966), no. 1, 36–45.
- [109] Adina Williams, Nikita Nangia, and Samuel Bowman, *A broad-coverage challenge corpus for sentence understanding through inference*, Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers) (New Orleans, Louisiana), Association for Computational Linguistics, June 2018, pp. 1112–1122.
- [110] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush, *Transformers: State-of-the-art natural language processing*, Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations (Online), Association for Computational Linguistics, October 2020, pp. 38–45.
- [111] Shijie Wu and Mark Dredze, *Beto, bentz, becas: The surprising cross-lingual effectiveness of BERT*, Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP) (Hong Kong, China), Association for Computational Linguistics, November 2019, pp. 833–844.
- [112] Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff

- Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Lukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean, *Google’s neural machine translation system: Bridging the gap between human and machine translation*, 2016.
- [113] Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel, *mT5: A massively multilingual pre-trained text-to-text transformer*, Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Online), Association for Computational Linguistics, June 2021, pp. 483–498.
- [114] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le, *Xlnet: Generalized autoregressive pretraining for language understanding*, Advances in neural information processing systems **32** (2019).
- [115] Andrea Zaninello and Alexandra Birch, *Multiword expression aware neural machine translation*, Proceedings of the Twelfth Language Resources and Evaluation Conference (Marseille, France), European Language Resources Association, May 2020, pp. 3816–3825 (English).
- [116] Wenhao Zhu, Hongyi Liu, Qingxiu Dong, Jingjing Xu, Lingpeng Kong, Jiajun Chen, Lei Li, and Shujian Huang, *Multilingual machine translation with large language models: Empirical results and analysis*, arXiv preprint arXiv:2304.04675 (2023).
- [117] Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler, *Aligning books and movies: Towards story-*

like visual explanations by watching movies and reading books, Proceedings of the IEEE international conference on computer vision, 2015, pp. 19–27.

Vita

Candidate's full name: Raghuraman Swaminathan

University attended (with dates and degrees obtained): Bachelor of Technology in Information Technology, Vellore Institute of Technology, 2020

Publications: Raghuraman Swaminathan and Paul Cook. 2023. *Token-level Identification of Multiword Expressions using Pre-trained Multilingual Language Models*. In Proceedings of the 19th Workshop on Multiword Expressions (MWE 2023), pages 1–6, Dubrovnik, Croatia. Association for Computational Linguistics.

Conference Presentations: 19th Workshop on Multiword Expressions (MWE 2023), Dubrovnik, Croatia, 2023.