UPPSALA
UNIVERSITET

Low-resource Semantic Role Labeling Through Improved Transfer Learning

Hannes Lindbäck

Uppsala University
Department of Linguistics and Philology
Master Programme in Language Technology
Master's Thesis in Language Technology, 30 ECTS credits

June 4, 2024

Supervisor:
Meriem Beloucif, Uppsala University

Abstract

For several more complex tasks, such as semantic role labeling (SRL), large annotated datasets are necessary. For smaller and lower-resource languages, these are not readily available. As a way to overcome this data bottleneck, this thesis investigates the possibilities of using transfer learning from a high-resource language to a low-resource language, and then perform zero-shot SRL on the low-resource language. We additionally investigate if the transfer-learning can be improved by freezing the parameters of a layer in the pre-trained model, leveraging the model to instead focus on learning the parameters of the layers necessary for the task. By training models in English and then evaluating on Spanish, Catalan, German and Chinese CoNLL-2009 data (Hajič et al., 2009), we find that transfer-learning zero-shot SRL can be an effective technique, and in certain cases outperform models trained on low amounts of data. We also find that the results improve when freezing parameters of the lower layers of the model, the layers of the model focused on surface tasks.

Contents

Preface

I would like to give a heartfelt thanks to my supervisor Meriem Beloucif. Without her ideas and support this thesis would without a doubt not have been finished on time, and most likely not at all.

1 Introduction

In the field of machine learning, everything is built upon the availability of data. Large language models such as OpenAI's GPT (Radford et al., 2018) or Meta's LLama (Touvron et al., 2023) require massive amounts of data for training. This data is however unsupervised, meaning that it does not need to be annotated in any way and is readily available from any text source. For other natural language processing tasks, supervised data is necessary. To train a model to recognize named entities, a dataset where each word is annotated with its entity (For instance CoNLL-2002 (Tjong Kim Sang, 2002)) is required. For a task such as part-of-speech tagging, a dataset of tokens tagged with the part-of-speech is needed. As the datasets for such supervised tasks need to be created, they are often limited in size and not available for all languages. Many times, the more rigorous the effort of creating the dataset is, the fewer languages the dataset is available in. The task of semantic role labeling is such a task where the laborious annotation efforts make the datasets more of an issue to create and therefore create a lack of datasets for smaller, more low-resource languages.

The goal of semantic role labeling (SRL) is to annotate each token in an input sentence with a label specifying the semantic role that the entity that the token encodes takes in the clause. For instance, in the sentence *The cat hunts the mouse*, *the cat* takes on the role of `agent`, the entity voluntarily performing the action, while *the mouse* takes on the role of `patient`, the entity involuntarily being affected by the action.

Datasets used for training models (Baker et al., 1998; Hajič et al., 2009; Palmer et al., 2004) to perform semantic role labeling are typically manually annotated, or induced from a manually annotated source. For this reason, SRL datasets are not available for many languages. Several exist for English, but for other languages, the options are much more limited. Non-English SRL datasets include the multilingual dataset CoNLL-2009 (Hajič et al., 2009), which is available for the (high-resource) languages German, Spanish, Catalan, Chinese and Czech; Ha et al. (2014), a Vietnamese version of Propbank (Palmer et al., 2004) with 5000 sentences and Bruton and Beloucif (2023), a semantic role labeling dataset for Galician with 5000 sentences.

Having evolved from feature-based statistical parsers, semantic role labeling is today often performed with the use of neural models (Jurafsky and Martin, 2023). With large neural language models, such as BERT (Devlin et al., 2019), possibilities of bypassing the data bottlenecks have emerged. By pre-training language models on multiple languages, one can get a multilingual model capable of transferring its knowledge in a task to another language that has not seen data of the task it is to learn. Utilizing this technique of transfer learning make it possible to be able to increase the performance of semantic role labeling models for low-resource languages for which there does not exist much data.

1.1 Purpose

The purpose of this thesis is therefore to investigate transfer learning in semantic role labeling in a multilingual context[1]. In our experiments, we use the multilingual CoNLL-

[1] Code available at: https://github.com/HannesLindback/Transfer-learning-SRL

2009 (Hajič et al., 2009) dataset to fine-tune a multilingual BERT model (mBERT on English SRL data and then transfer the learned embeddings to a downstream language (either Spanish, Catalan, German or Chinese) by zero-shot label predictions. We also try freezing parameters in the pre-trained model as this has been shown to increase performance for transfer learning to low-resource languages (Chowdhury et al., 2022; Zoph et al., 2016) by inducing the model to only fine-tune parameters that are of importance to the task. The process of our experiments is therefore:

1) Fine-tune mBERT on the English dataset of the CoNLL-2009 shared task.

2) Freeze parameters of mBERT layerwise.

3) Evaluate the model on CoNLL-2009 data in Spanish, Catalan, German and Chinese.

Based on this, the research questions this thesis will attempt to answer are:

1) Can transfer learning be utilized to overcome the lack of data for low-resource languages for semantic role labeling?

2) Can the transfer learning be improved by freezing parameters in the model?

In regards to the second question, we hypothesize that the parameters that will cause an improvement in the performance of the model when frozen are located in the layers where the average attention weight for SRL is high. Previous studies in large language models have indicated that certain models have higher attention weights than other layers for certain tasks and are as such more specialized to that task (Clark et al., 2019; Jawahar et al., 2019; Kovaleva et al., 2019). We hypothesize that since the layers with high attention weights for SRL are already capable of the task, the model will benefit from training the layers not as capable of semantic role labeling instead of the layers already proficient in the task.

2 Background

2.1 Semantic Roles

Semantic parsing is within the field of natural language processing the task of annotating a sequence of text with labels corresponding to some semantic feature. Such semantic features that can constitute a task within semantic parsing are for instance semantic role labeling and coreference resolution. In semantic role labeling (SRL), the task is to annotate the input sequence with labels of the semantic roles of the arguments in the sequence. As an example, consider the sentence "The woman loves the man". In this sentence, the predicate "loves" takes the arguments *the man* and *the woman*. These arguments constitute the relations subject and direct object. In a simple clause, the subject is the argument that the predicate, the finite verb, modifies and agrees with. Consider for instance the clause "The cat eats the birds", where the verb agrees in number with the singular cat, the subject, and not the plural birds, the direct object. It is important to note however that these operate on the grammatical level. Semantically, i.e. on the meaning level of the clause, these arguments are instead called agent and patient, with the agent being the argument that enacts the action, and the patient being the one that is affected by the action taken by the agent. While in this example the semantic roles correspond well with the grammatical relation of the clause, this is often not the case. Take for instance the clause "the man is loved by the woman". Now the voice has changed from the active to the passive, changing also the grammatical makeup of the clause, reducing the saliency of the agent by demoting it grammatically from the subject to an oblique marker by putting it behind the preposition "by". Likewise, the patient *the woman* has been promoted to the subject of the clause. But while the grammatical roles of *the man* and *the woman* have changed, their semantic roles are still the same as the semantic content of the clause is (more or less) unchanged.

$$\textit{The woman loves the man} \approx \textit{The man is loved by the woman}$$

This is what constitutes the difference between semantic roles and grammatical arguments, and what makes the task of labeling semantic roles different from other types of syntactic annotation tasks.

Table 1 lists some of the more commonly defined semantic roles. The list is by no means complete as there are many different theories and definitions regarding what constitutes a specific semantic role. This is even further complicated by the sometimes rather diffuse relationship between semantics and syntax, as previously mentioned: a specific semantic role is by no means limited to one specific syntactic usage. As demonstrated above, the patient can be both subject and direct object, and the agent can be both subject and adverbial. This extends also to the *instrument* role. In the sentence "John broke the window with a rock", "rock" is used by the agent to perform the action: it is the instrument and is suitably also grammatically an adverbial. However, in the sentence "The rock broke the window", the instrument is now syntactically the subject of the sentence. This type of syntactical alternation for semantic roles is known as diathesis alternation (Jurafsky and Martin, 2023) and occurs for several roles for many verbs. The dative role (a non-volitional, but still conscious participant), may be both the subject and direct object (Givón, 2001):

Role	Definition
Agent	The volitional causer of an event
Patient	The participant that is most affected by an event and registers a state-of-change as a result of the event.
Dative	The non-volitional causer of an event.
Instrument	A participant that is used by the agent to perform the action.
Manner	The manner in which an event occurs.
Benefactive	The participant for whom the event was performed and who experiences the event.
Result	The end product of an event.

Table 2.1: Some commonly defined semantic roles. Adapted from Givón (2001) and Jurafsky and Martin (2023).

Dative subject: "They heard the music"
Dative object: "He amused them"

The ways in which these alternations occur can be divided into different verbal classes. Levin (1993) classifies about 3100 English verbs depending on the meaning that the verbs convey and the type of diathesis alternation that they show. Givón (2001) does a simpler, more linguistically universal classification of verb alternation based on the concept of transitivity, where an intransitive verb will have a subject that either has the role agent, patient or dative, and a transitive verb will have a subject that is an agent and a direct object that is a patient. However, as demonstrated, this does not hold for all verbs, but since this pattern tends to be the norm. It is possible to say that this is the prototypical way that these semantic roles appear. A proto-agent therefore will be more likely to be deliberate, active and volitional, whereas a proto-patient will be more likely to be concrete, physically affected and undergoing a change-of-state (Givón, 2001; Jurafsky and Martin, 2023). The first example given "The woman loves the man" is therefore not very prototypical as the verb *loves* has an agent that is less volitional (to fall in love is seldom a deliberate choice) and an agent that is more mentally than physically affected. A better example of prototypical agents and patients can instead be found in the sentence "The woman murdered the man". These examples are nonetheless only prototypes, even though a majority might conform to it, some will always fit less well: "In principle, thus, if one probes deep enough, each verb defines its own unique propositional frame, thus its own unique array of semantic roles." (Givón, 2001). This is reflected in the works of FrameNet (Baker et al., 1998), and to a lesser extent PropBank (Palmer et al., 2004).

2.1.1 PropBank and FrameNet

PropBank (Palmer et al., 2004) and FrameNet (Baker et al., 1998) are two early semantically annotated resources on which many later projects are based. PropBank is structured around the use of prototypical semantic roles in combination with the idea that each verb defines its own set of unique roles. To handle this in annotation PropBank uses numbered labels: Arg0, Arg1, Arg2, etc., with the first two being the roles agent and patient in the prototypical sense, and any higher numbered labels being any other semantic roles applicable to that specific verb. This enables PropBank to have an annotation system that does not have to follow one specific definition of the semantic roles but rather keep the system more functional and operable across different languages. FrameNet on the other hand does not take prototypical roles into consideration and is solely focused on creating definitions of roles based on the specific semantics of the verb, or in this case a collection of verbs (and nouns) that have a similar semantic definition. One collection of semantically similar verbs constitutes one frame. For each frame, there are specifically defined roles that precisely describe the semantic functioning of the different arguments.

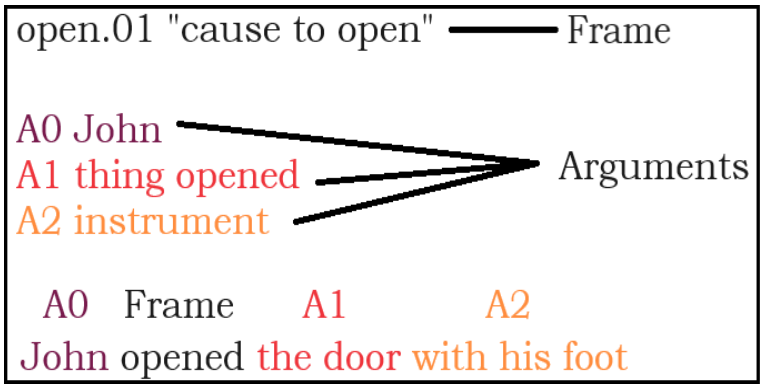

Figure 2.1: A verbal frame *open* with the sense "cause to open", with three numbered arguments. A0 and A1 in this case roughly correspond to agent and patient.

2.2 Large language models

Since the earliest occurrences of pre-trained embeddings with systems such as GLoVe (Pennington et al., 2014) and SkipGram models (Mikolov et al., 2013), pre-trained language systems have been important for the field of Natural Language Processing, and even more so since the emergence of pre-trained language models with contextualized embeddings. While the first large language model of this kind, ELMo (Peters et al., 2018), was based on LSTM-architecture, later models use the transformer architecture (Vaswani et al., 2017). Examples of such transformer-based large language models include GPT (Radford et al., 2018) and BERT (Devlin et al., 2019), and more recently also models such as META's Llama (Touvron et al., 2023). These transformer-based language models have since their first appearances been able to continuously break the current state-of-the-art on various natural language understanding tasks. The method that has enabled this is the pre-training of the model on a general language task, and then fine-tuning the model on task-specific data for whichever downstream task is required. GPT models are for instance pre-trained on continuously predicting the next token in a sequence, BERT-based models are instead pre-trained on the so-called Masked Language Model task and Next Sentence Prediction, which is described in further detail below.

2.2.1 Transformers

The transformer architecture (Vaswani et al., 2017) is a network largely based on attention: a technique that enables the model to take long-distance contextual dependencies into account when creating the encoding for a token. An early version of attention, additive attention (Bahdanau et al., 2015), generates contextualized attention vectors for each token in a sequence by summing an annotation vector, containing information about the previous and next tokens, with the previous hidden state. This enables the model to create a better contextual representation of each token processed. However, as the network is recurrent, that is, it processes each token in one direction at a time, some contextual information about tokens further away from the current token will still be lost. To account for this, the transformer does away with recurrence altogether, and instead focuses solely on self-attention to create the token encoding. The attention used in the transformer scaled dot-product attention (Vaswani et al., 2017) (called so due to the mathematical operations used in the attention calculations). In scaled dot-product attention, a contextual representation of a token is generated by a series of matrix operations: from the input $\vec{x}$ a query $\vec{q}$, a key $\vec{k}$ and a value $\vec{v}$ is procured from the matrices Q, K and V. Next, softmax is applied on the scaled dot product of $\vec{q}$

and $\vec{k}$, which is then weighted with the value vector $\vec{v}$ to generate the attention score z of the token:

$$z = softmax(\frac{\vec{q} \cdot \vec{k}}{\sqrt{d_k}})\vec{v}$$

Through these operations, the attention score z will be a weighted average of the other tokens in the sequence, or more intuitively, a representation of the contextual importance of the other tokens for x.

A second important component of the attention that is used in transformers is the multi-headed attention. Instead of doing one single scaled dot-product calculation for each input sequence, the inputs are split across N different linear projections, which are then processed in parallel. The reason for this is that it enables the model to attend with different weights to different parts of the input sequence. Instead of having the attention be focused on one single part of the sequence, e.g. the previous token, the model can have one attention head focused on that, and another attending to e.g. the next token, or tokens that coreferent with current token. The different attention heads' weights are then concatenated back together to form one attention score. Taken altogether, projecting the attention over multiple heads allows the model to create an even better representation of the input, and, as the calculations are done in parallel, no extra cost is required.

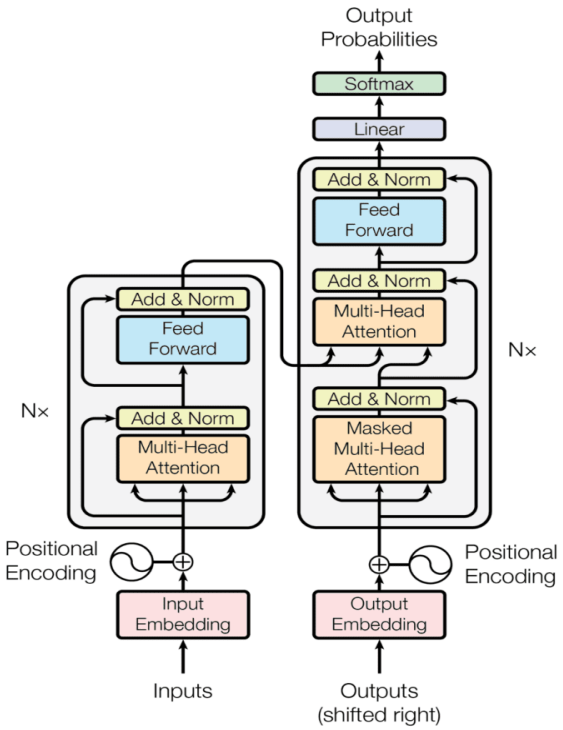

Figure 2.2: A transformer encoder-decoder network. Image from Vaswani et al. (2017).

The original "vanilla" transformer as presented in Vaswani et al. (2017) has an encoder-decoder structure (illustrated in Figure 1), where the inputs are given to the encoder, to then be used by the decoder to generate the outputs. But before an input sequence is passed to the model, the tokens are first mapped to a word embedding and given a positional encoding. The reason for the positional encoder is that as the transformer does not use recurrence, i.e. it processes the entire sequence at once and does not go through the sequence one token at a time, it needs some way to take the word order of a sequence into account. In the encoder block, the inputs then go through

the first attention module and the fully connected non-linear feed-forward network in addition to layer normalization modules. The decoder block of the transformer is almost identical to the encoder, except that there are two attention modules which both have slightly different functions. In the first attention module of the decoder, the tokens after the current token being processed are masked in order to prevent the model from "cheating" by looking at the next token. In the second attention module, the query comes from the input to the decoder and the key and value from the encoder block. This enables the module to also take into account the input from which it is supposed to generate the output. Not all transformer-based models utilize both the encoder and decoder part of the network, however. GPT-like models utilize only the decoder (Radford et al., 2018), and BERT uses only the encoder part of the network (Devlin et al., 2019).

2.2.2 Pre-training of large language models

The pre-training strategies are what have set the large language models apart from earlier models. By pre-training the models on massive amounts of data the models are able to get a strong base understanding of language. This understanding can then be augmented further by fine-tuning the model on smaller, task-specific data. The pre-training of BERT consists of two tasks, the Masked Language Model (MLM) and Next Sentence Prediction (NSP). In MLM, the task is to predict the next token in a sequence, but in order to enhance the bidirectionality of the model, the token to be predicted is masked 15% of the time. 80% of the time that it is masked, the token is replaced by the special token [MASK], 10% of the time it is replaced with a random token and for the remaining 10% the token is left unchanged. In the task NSP, the model is given two sequences, separated by the special token [SEP]. The goal of this task is for the model to predict if the two sequences are contiguous or not (Devlin et al., 2019).

The main idea behind large pre-trained language models such as BERT is to utilize the massive amounts of information stored in the embeddings of the pre-trained model, and then specialize this information on some downstream task by fine-tuning on new, typically smaller amounts, of data. Such applications are referred to as transfer learning (Raffel et al., 2020), as the pre-trained information is transferred to a new task to improve the performance. That is, the embeddings used by the model are not trained from scratch, but rather modified slightly to fit the new data. However, in certain cases of transfer learning by fine-tuning on new data, it might be advantageous to not modify all embeddings of the pre-trained model. For instance, in cases where one has only access to a very small amount of data that would not be enough to fit the model completely, it can be beneficial to freeze pre-trained parameters that are beneficial to the downstream task so that these are not updated during training. This technique has previously been utilized for machine translation of low-resource languages to overcome the lack of data for such languages. Such experiments have been conducted in settings where a high-resource language pair is used to learn the embeddings of a target language, which is then used as target for a low-resource source language. When fine-tuning with low-resource data, the already learned target language embeddings are frozen and only the low-resource language embeddings are updated. Using this setup, Zoph et al. (2016) managed to increase the BLEU-score of the low-resource languages Hausa, Uzbek, Urdu and Turkish, with the use of French-English as the high-resource pair from which the parameters were transferred. Chowdhury et al. (2022) utilized the transfer learning freezing technique with transformer models, where they froze the layers in either the encoder or the decoder. Freezing parameters in

the encoder showed an increase in performance, whereas freezing parameters in the decoder instead lowered the results of the experiments.

2.3 Related work

2.3.1 Semantic Role Labeling

Modern approaches to computationally label semantic roles are often based on neural networks. For neural semantic role labeling a common way is to use methods of general sequence labeling. Given a sequence, calculate the highest probable tag sequence, based on a dataset with annotated examples. This is often done by encoding each input token with pre-trained embeddings of some kind, which is transformed into a probability distribution of the most probable tag for each token with a multi-layered perceptron and a softmax function. Shi and Lin (2019) used the pre-trained large language model BERT to encode each input token in a sequence. The encoded token was then concatenated with an embedding indicating whether or not the token is a predicate. This was then put through a one-layered MLP to get the probability distribution for the most likely tag sequence for that input sequence. Other methods include using bidirectional RNN models, either with attention (Tan et al., 2018) or without (Zhou and Xu, 2015), or BiLSTM models (He et al., 2017).

2.3.2 BERTology

The novelty of the attention mechanism and the new ways it, and pre-training schemas make large language models handle and understand language have brought about numerous studies, deeply investigating the functioning of the self-attention mechanism of pre-trained language models. Since BERT was one of the earliest pre-trained transformer models, and since it had such a large impact, there are plenty of studies specifically investigating the layers and attention heads of BERT, to such an extent that there almost came about the subfield of "BERTology" (Rogers et al., 2020).

BERT's layer-wise functioning

The general way that language models deal with language representation on a layer-wise basis is that earlier layers handle more basic linguistic features on a phrasal level, such as basic word classes and part-of-speech. Next, in the middle layers are the representations of more complex syntactic features, with clause-level features such as dependency and constituency structures. Last are semantic features handled, such as coreference mentions, named entities and semantic roles (Tenney, Xia, et al., 2019). This same pattern can be found in BERT as well. Using the probing tasks of Conneau et al. (2018), Jawahar et al. (2019) investigate where in the layers of BERT separate linguistic functions are handled by using the output of one single layer as the output of a classifier for each of the probing tasks. The results of the probes confirm this classic pipeline, as the five earliest layers of BERT achieve the highest scores of surface tasks, such as sentence length and words' presence in sentences. Classifiers probing for syntactic features get their highest scores in layers five to nine, and semantic probes achieve their best results mostly in the deepest layers. However, in neither of the tasks is there one specific layer that achieves outstanding results. Several layers instead achieve similar results. This indicates that not one specific layer is solely responsible for one single task, but rather that multiple layers appear to have similar functions. This is supported by the results of Tenney, Das, et al. (2019), where they found that weights relating to a single task were spread out across multiple layers, and

not localized to one single layer. For semantic tasks, this observation was especially strong, with weights sometimes almost uniform across all layers of the model meaning that the entire model was involved in the processing of semantic features. These results hold as well for BERT in non-English languages: both Dutch monolingual BERT and multilingual BERT do not localize the weights for linguistic tasks in a single layer but rather use multiple layers in tandem. A key difference between monolingual and multilingual BERT in this aspect is that for the multilingual model, the important layers tend to be situated earlier than in the monolingual versions (de Vries et al., 2020).

BERT's head-wise functioning

The multi-headedness of the transformer's attention modules enables the model to not only have the layers focusing on the different linguistic features but also have individual attention heads attending more, or less, to different parts of language. As explained in Section **1.2.1**, for any attention head in the model, attention maps for any input sequence $s = [s_1, ..., s_n]$ of length n are created by taking the normalized dot product of query and key vectors $\vec{q}$ and $\vec{k}$. This produces a matrix A of shape $n * n$ where the weight to a token A_i from any other token A_j is a measurement of the importance of A_j for the contextual representation of A_i.

In many ways the attention heads follow the classic NLP pipeline evident in the layers of BERT: in attention heads in the earlier layers of the model there is a higher tendency for the heads to have higher entropy than in the deeper layers, that is, they show broad attention patterns, distributing the weights more evenly across all tokens in the sequence (Clark et al., 2019). This is consistent with the claims of Jawahar et al. (2019) and Tenney, Das, et al. (2019) that earlier layers are more focused on surface-level features in a sequence and that attention heads in the later layers are more focused on syntax. A probe on individual attention heads for separate syntactic dependency relations show that single heads in layers 4, 6, 7, 8 and 9 specialize in specific syntactic relations, but that no single head does well on syntax overall (Clark et al., 2019). Investigations in individual attention heads' semantic representations are not as numerous as the syntactical studies, but tentative support for the claim of Tenney, Das, et al. (2019), that semantics is spread out across the layers of the model, can still be found. Kovaleva et al. (2019) average the maximum attention weights for arguments in a semantic relation for all attention heads for 473 sentences, suggesting that one head in layer 1 and one head in layer 7 are especially attentive to semantic roles. A probe for coreference resolution identified head 4 in layer 5 is especially capable of handling this type of semantic relation (Clark et al., 2019).

Adjustments of multi-head attention

There have been several studies pointing out flaws with the multi-head attention mechanism. One such flaw is regarding the size of the model, or rather, the number of heads in the attention module. The general tendency in neural natural language processing is that larger models with more parameters perform better than smaller models (Devlin et al., 2019; Kaplan et al., 2020). Despite this, investigations into the attention heads of BERT that the model appears to be over-parameterized, and that removing or disabling heads in the model can lead to improvements in performance. Voita et al. (2019) used layer-wise relevance propagation (Ding et al., 2017) to evaluate the relevance of individual attention heads for the model's prediction in a translation task, showing that there are only a few heads (specialized in word order, syntax and infrequent words) that appear to be important for the model to accurately perform

the task, a majority of the non-important heads can be pruned without severely affecting the model's performance. Michel et al. (2019) found that it was possible to prune 20% of the heads in the WMT model of Vaswani et al. (2017) without affecting performance, and up to 40% of the heads in BERT could be pruned without registering a noticeable performance drop for the MultiNLI task (Williams et al., 2018). Kovaleva et al. (2019) swapped the weights in all attention heads in a layer at a time to constant weights and noticed for some layers and tasks an improvement compared to using the learned attention weights. Hassid et al. (2022) discovered that replacing half the learned attention weights with constant weights caused no major drop in performance across several tasks.

Attempts have also been made to improve the attention mechanism in more advanced ways than just removing heads. Wang et al. (2022) noticed that disproportionately large amounts of attention are given to the two special tokens [CLS] and [SEP]. In an attempt to correct this, they attempt to guide the attention of the model to distribute more evenly across the important tokens of an input sequence by using discriminating models to produce a probability distribution over the attention weights and to guide the weights in an attention matrix to be more dissimilar to each other. Deshpande and Narasimhan (2020) create an attention guiding schema for improving the distribution of the attention weights while pre-training the model by adding a regularization term to force the weights into five predetermined patterns.

3 Methodology

The main research task of this project is to experiment with performing SRL on downstream languages through transfer learning, and to experiment with freezing parameters in a pre-trained large language model in order to investigate whether this can improve the transfer of semantic knowledge to the downstream language. More concretely, we test if freezing the parameters of a layer has a positive effect on the result when doing zero-shot semantic role labeling. Two types of experiments are performed:

`Monolingual`: As a baseline, we train and evaluate a model on the same language, with varying amounts of data to simulate a low-resource setting.

`Transfer learning`: In this setting, we train the model on English data and evaluate it on either Spanish, Catalan, German or Chinese data. During training, the parameters of the pre-trained model are frozen layer-by-layer. Ranging from no layers frozen, to more than half of the layers frozen.

3.1 Data

The data used in the experiments is from the CoNLL-2009 dataset (Hajič et al., 2009) - a multilingual dataset in the languages: English, Spanish, Catalan, Chinese (Mandarin), German and Czech. For all languages, the dataset uses an extended version of the CoNLL-U format [1], with each token annotated with syntactic information (part-of-speech, morphological features and dependency relations) and with the following semantic information:

- if the token is an argument-bearing predicate,
- predicate sense identifier,
- argument label, indicating to which predicate the token is an argument, and what kind of argument it is.

The semantic portion of the CoNLL-2009 dataset is based on frame semantics (see Section 2.1.1), meaning that every predicate in the dataset is a frame, with a specific sense described and with accompanying arguments and examples. For English, such argument-bearing predicates can be both nominal and verbal. For the other languages, predicates are verbal only. Apart from examples illustrating the sense of the frame, the frame also specifies which arguments the predicate can take, and the semantic meaning of each argument.

The tag sets for the argument labels vary slightly across the languages. For English and German, the argument labels are either numbered arguments (A0, A1, A2, A3, A4, A5) or "AM"-arguments, encoding some further adverbial sense such as temporal, locative or manner properties. The tag set for the Chinese data consists of numbered arguments A0-A5 together with a high number of adjunctive arguments. For Catalan and Spanish, there are four numbered arguments for verbs and two argument types for adjuncts. Each argument also has its semantic role described in the label (agent, patient, etc.). The Czech tag system is completely different from the other languages

[1] https://universaldependencies.org/format.html

in the dataset in the sense that it contains no numbered arguments as is otherwise the norm in frame semantics. It instead just uses abbreviations of semantic roles (ACT for actor, PAT for patient, etc.).

The are also differences across the languages regarding the amount of predicates in the dataset as a whole and in a single sentence. As can be seen in Table 3.1, German has a very low number of predicates compared to the other languages: only 2.7 percent of the tokens in the German data are predicates, and there are no more than five predicates in a single sentence. In comparison, Chinese has got an extremely high amount of predicates found in a single sentence.

Another quirk with the German data is the imbalance between the train and evaluation datasets. For the other languages, there is a similar number of sentences with at least one predicate in both the train and evaluation sets. For German the test dataset contains significantly fewer sentences with at least one predicate. As the test dataset is already small in size, this means that there are very few examples per label compared to the rest of the languages. Due to this a new train/test split was used: the original train/test sets were combined, and then randomly split with 5% of the sentences to the test set and 95 % to the train set. This made for a more even distribution of sentences with predicates, making the data more suitable for analysis.

The annotation system used in the experiments consists of predicate labels and argument labels. A predicate label consists of $p + predicate_n$ based on the order that the predicates appear in the sentence. So that the first predicate in a sentence gets the label *p0*, the second *p1* and so on. The argument labels consist of the argument's predicate's label + A*n*. For example, the Ao argument of the third predicate in a sentence would be labeled as *p2Ao*. During pilot testing of the experiments, the accuracy scores for the argument labels were very low. This was assumed to be because of the complex hierarchical label structures created by the high amounts of predicates in a sentence. For example, in Chinese, there are 39 possible Ao labels as there can be up to 39 predicates in a sentence. A very low base accuracy means that it gets more difficult to draw conclusions and analyze the results, which makes this thesis less valuable. As the goal of this project is not to compete with the CoNLL-2009 leaderboard, but rather to analyze the results to investigate a new technique for zero-shot SRL, we limit the maximum number of predicates in a training sentence to five. The evaluation sentences are kept as-is. For similar reasons only numbered argument labels are used in both the train and test data.

	German	new_German	English	Spanish	Catalan	Chinese	Czech
Train data size (sentences)	36020	36119	39279	14329	13200	22277	38727
Evaluation data size (sentences)	2000	1901	2399	1725	1862	2556	4213
Predicates (%)	2.7	2.7	18.7	10.3	9.6	16.9	63.5
Max number of pred. in a sent.	5	5	16	15	15	39	N/A
One pred. per sent. in eval. (%)	26	38	92	95	96	94	N/A
One pred. per sent. in train (%)	39	39	90	96	97	94	N/A

Table 3.1: The languages of CoNLL-2009. new_German = the balanced train/test split of the German data.

3.2 Model selection

The work in this thesis utilizes multilingual BERT (Devlin et al., 2019) as a pre-trained model for predicting semantic role labels. At the time of writing, the BERT models were first published more than five years ago - ancient, by the standards of how fast the field of NLP develops. Given that there are plenty of more recent and better-performing

language models released, I wanted to give some reasoning for why such an outdated model was chosen. The reasons are both practical and scientific. One advantage BERT has over the best language models released to date is that the training data is known and controlled. For modern language models such as Llama, Mistral, GPT>1, the training data has not been released. As these models are trained on trillions of tokens possibly sourced from crawling the web, it is not possible to control that the models have not already seen parts of the test data, potentially spoiling the experiments. With BERT the training data is known. Monolingual BERT was trained on the BookCorpus (Zhu et al., 2015) dataset and the English Wikipedia, while the multilingual BERT model is trained on Wikipedia pages in each language. The data is known, ensuring no data contamination. BERT is also suitable due to practical reasons: the experiments were performed on limited compute resources, making BERT's smaller size an advantage over more recent billion-parameter models. These reasons in combination with that BERT still performs relatively well for its size made it a suitable choice for this thesis's experiments.

3.3 Training setup

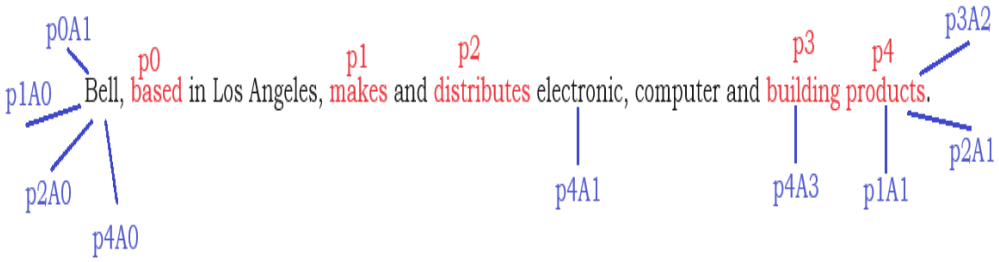

Figure 3.1: A sentence from the English CoNLL-2009 dataset with multiple labels on single tokens.
Red tokens and labels: predicates, numbered after their appearance in the sentence. Blue tokens and labels: numbered arguments, labeled together with their predicates.

The primary task that the experiments performed in this work are to investigate, is to see the effect it can have to freeze layers in a pre-trained large language model when performing zero-shot SRL on a downstream language. In the CoNLL-dataset, a single token can have multiple labels, as illustrated by the example sentence in Figure 3.1, where the first word of the sentence, *Bell*, is the argument of all four predicates in the sentence. As such, the setup for semantic role labeling is handled as a multi-label classification problem. Given a dataset D with pairs of input tokens x and true label vectors y where:

> x represents a token in D,
> y represents one-hot-encoded vectors of size n *labels*, where each element represents a unique predicate or argument label and is 1 if that label is present and 0 if otherwise,

the task of the model is to predict a label vector $\hat{y}_i$ for each token x_i in D. The label vector y is one-hot, meaning that every label takes on a binary encoding. Because of this the problem of classifying one token x_i is treated as multiple binary classification problems, where the model is to predict 1 or 0 for each label in $\hat{y}_i$. For this binary cross entropy (eq. 1) is used as the objective function to make the model learn the mapping

$f : x \rightarrow y$ such that the predicted label vectors $\hat{y}_i$ are as close as possible to the true label vectors y_i for each token x_i in D.

$$\text{Binary Cross Entropy}(y, \hat{y}) = -\frac{1}{N}\sum_{i=1}^{N}[y_i \log(\hat{y}_i) + (1-_i)\log(1-\hat{y}_i)]$$

During training, both experiments use a pre-trained BERT model to create a contextual representation of each input token from the training data. A one-layer multi-layered perceptron is used to transform the output of the pre-trained model to the tag space. A sigmoid activation function is then used on the predicted vector to get the final predicted labels for the input. For the `Monolingual` experiments no parameters are frozen. For the `Transfer learning` experiments parameters are frozen layerwise in the pre-trained model. All experiments are performed using the hyperparameters in Table 3.2. The models used in the experiments are trained on a Nvidia T4 GPU until convergence.

Hyperparameter	Value
Learning rate	1e-05
Dropout	0.5
Weight decay	5e-04
Batch size	64

Table 3.2: Hyperparameters used during model training.

The models are evaluated using F-score, the harmonic mean of precision P and recall R:
$P = \frac{TP}{TP+FP}$ $R = \frac{TP}{TP+FN}$ $F = \frac{2\times P\times R}{P+R}$
Where:

TP: The true positives. The correctly predicted positive examples.
FP: The false positives. The incorrectly predicted positive examples.
FN: The false negatives. The incorrectly predicted negative examples.

3.4 Experimental setup

Two types of experiments are performed and evaluated. For the first experiments, `Monolingual` a model is trained and evaluated on data in the same language, with no parameters frozen, but with varying amounts of data to simulate a low resource setting. Either the full train data amount is used, or subsets of the dataset with 50% or 5% of the data (see Table 3.3). As the sizes of the training datasets differ across the languages, this too means that the subsets will be of different sizes for the different languages. For Spanish and Catalan, the smallest subset becomes very small with only a couple of hundred sentences for training.

	new_German	English	Spanish	Catalan	Chinese
Full train data size (sentences)	36119	39279	14329	13200	22277
50% train data size (sentences)	18060	19640	7165	6600	11139
5% train data size (sentences)	1806	1964	716	660	1140

Table 3.3: The sizes of the different subsets of training sets used in the experiments. new_German = the balanced train/test split of the German data.

In the second experimental setting `Transfer learning`, the model is trained and evaluated on different language pairs utilizing frozen parameters to improve the transfer learning of the downstream language. The models are always trained on English data and then evaluated on either German, Chinese, Spanish or Catalan data. Due to the different annotation systems of the Czech data, it is not included in the experiments as it would have been a too difficult and time-consuming task to translate the English argument label type to the Czech label type (see Section 3.1.). The experiments started by freezing one layer at a time. As the first four layers, seemed to cause the highest increase in F_1-score, we reran the experiments with the frozen layer combinations 1, 2, 3, 4; 1, 2, 3 and 1, 3, 4.

In addition to these two experimental settings, we also created heatmaps of the attention weights in the individual attention heads when performing SRL on the dataset. These heatmaps were created with the following method after (Kovaleva et al., 2019):

- For each sentence in the development datasets of Spanish, Catalan, German and Chinese, extract pairs of predicate-argument tokens. That is, the tokens that either are predicates or arguments of predicates. As some attention heads are attuned to the next/previous tokens, only predicate-argument pairs that are not contiguous are extracted.
- For each predicate-argument pair in a sequence, choose the pair with the highest attention weight and calculate the average attention weight per sentence.
- Repeat for each attention head in the model.

These heatmaps indicated that the heads in layers 5-10, and especially layers 5 and 6, had the highest attention weights for SRL. We therefore also freeze the parameters in layers 5 and 6, and in layers 5, 6, 7, 8, 9 and 10.

4 Results

The purpose of this study is to investigate whether it is possible to improve the transfer learning of a semantic role labeling model by freezing parameters in the model's layers. This can then be used to improve SRL for a low-resource language by fine-tuning the model on data in a high-resource language with parameters frozen, and then perform zero-shot SRL on the low-resource language. These experiments are performed on four languages: Spanish, German, Catalan and Chinese. In the `Transfer learning` setting (the transfer learning models), five combinations of layers are frozen, in addition to the **baseline**: zero-shot SRL with no frozen parameters. These are compared to the `Monolingual` models, which are trained with varying amounts of data in order to simulate a low-resource setting. In all tables below, the *p*-labels are predicate labels, while the *A*-labels are the score of each argument label for all predicates. Only labels with F_1-scores above 0.0 are included in the tables.

4.1 Spanish

Training setup	Model	Total	p0	p1	p2	p3	p4	A0	A1	A2
	English-Spanish **baseline**	0.79	0.72	0.54	0.35	0.1	0.0	0.04	0.04	0.0
`Transfer learning`	English-Spanish [1, 3, 4]	0.81	0.78	0.62	0.49	0.4	0.05	0.1	0.09	0.02
	English-Spanish [1, 2, 3]	0.8	0.75	0.57	0.42	0.35	0.16	0.1	0.09	0.02
	English-Spanish [1, 2, 3, 4]	0.8	0.77	0.59	0.45	0.39	0.13	0.1	0.09	0.03
	English-Spanish [6, 7]	0.8	0.75	0.58	0.45	0.2	0.0	0.08	0.07	0.01
	English-Spanish [5, 6, 7, 8, 9, 10]	0.78	0.67	0.47	0.31	0.05	0.0	0.05	0.03	0.0
`Monolingual`	Spanish **baseline**	0.95	0.98	0.94	0.91	0.87	0.8	0.36	0.3	0.26
	Spanish 50%	0.94	0.97	0.93	0.88	0.82	0.75	0.31	0.27	0.2
	Spanish 5%	0.89	0.96	0.87	0.68	0.12	0.0	0.11	0.09	0.01

Table 4.1: F_1-scores for the models trained on Spanish data with the two training setups `Transfer learning` and `Monolingual`.
`Transfer learning`: Trained on Spanish data, evaluated on Spanish data. **baseline**: No frozen parameters. For the other models in `Transfer learning` the number in [] indicate the layers that were frozen during training.
`Monolingual`: Trained on Spanish data, evaluated on Spanish data, with no frozen parameters. Model **baseline**: trained on the full data amount. The other two `Monolingual` models trained on 50% or 5% of the data amount.
All values are the average of five models' result.

Table 4.1 shows the result of the `Transfer learning` and `Monolingual` settings when trained on Spanish data. Unsurprisingly, the `monolingual`model trained on the full data amount achieves the best F_1-scores across all predicate and argument labels. For all settings, the models struggled with correctly labeling arguments, as these scores are considerably worse than the predicate label F_1-scores. In the `Monolingual` setting, training the model on half of the data had a relatively small impact on the label scores, while the low-resource simulation with only 5% of the data had a severe impact on the F_1-scores of the less common labels (p>2 and all argument labels). In the `Transfer learning` setting the model with the parameters in layers 5-10 frozen achieved the worst scores, and the models with the parameters in layers [1, 3, 4] and in layers [1, 2, 3, 4] achieved the highest F_1-scores. Apparently, freezing [1, 3, 4] improved the models

Training setup	Total	p0	p1	p2	p3	p4	A0	A1	A2
English-Spanish [0]	0.79	0.72	0.54	0.35	0.1	0.0	0.04	0.04	0.0
English-Spanish [1]	0.81	0.77	0.61	0.49	0.36	0.06	0.09	0.08	0.01
English-Spanish [2]	0.8	0.78	0.61	0.48	0.36	0.03	0.09	0.08	0.01
English-Spanish [3]	0.8	0.77	0.6	0.48	0.36	0.03	0.09	0.08	0.01
English-Spanish [4]	0.8	0.75	0.59	0.47	0.34	0.0	0.09	0.07	0.01
English-Spanish [5]	0.8	0.75	0.59	0.45	0.29	0.0	0.08	0.07	0.0
English-Spanish [6]	0.8	0.75	0.58	0.45	0.32	0.0	0.08	0.07	0.01
English-Spanish [7]	0.79	0.73	0.54	0.41	0.21	0.0	0.07	0.05	0.0
English-Spanish [8]	0.8	0.75	0.57	0.43	0.26	0.01	0.08	0.06	0.01
English-Spanish [9]	0.8	0.75	0.58	0.46	0.25	0.0	0.07	0.06	0.0
English-Spanish [10]	0.8	0.74	0.56	0.43	0.24	0.0	0.07	0.06	0.0
English-Spanish [11]	0.8	0.75	0.58	0.45	0.28	0.0	0.07	0.07	0.01
English-Spanish [12]	0.79	0.72	0.54	0.38	0.16	0.0	0.07	0.05	0.0

Table 4.2: F1-scores for the models trained on Spanish data with individually frozen layers. The numbers in [] indicate which layers that were frozen. [0] = No frozen layers. All values are the average of five models' results.

performances for labeling predicates, while freezing [1, 2, 3, 4] proved slightly better for labeling arguments. Both of these layer combinations outperformed the `Transfer learning` **baseline** where no parameters were frozen. They also performed better at the more infrequent labels compared to the low-resource `Monolingual` setting, as all arguments and predicates 3 and 4 more often got labeled correctly by the `Transfer learning` [1, 3, 4] and [1, 2, 3, 4] models than by the `Monolingual` model trained on 5% of the dataset.

Table 4.1 shows the results of models where the parameters were frozen one layer at a time. The best performing models where the ones with one of the first four layers frozen. The model with zero layers frozen achieved the worst F_1-score, thightly followed by the model with layer 12 frozen.

4.2 Catalan

The results of Catalan are quite similar to those of Spanish. The `Monolingual` model trained with the full data amount achieved the highest scores, although it tied with the model trained on 50% of the data for label p0 and only substantially outperformed that setting for the p4 label. Also, as with the Spanish model, the `Transfer learning` [5, 6, 7, 8, 9, 10] setting performed the worst of the models in setting `Transfer learning`, and models with frozen layers [1, 3, 4] and [1, 2, 3, 4] achieved the highest F_1-scores. However, in the case of Catalan it is the setting [1, 2, 3, 4] that performs best, rather than [1, 3, 4] as is the case for Spanish. One other difference from the Spanish results, is that the most of the models in the `Transfer learning` setting outperformed (except for label p0) the low-resource `Monolingual` model, trained with 5% of the data. Even the worst performing `Transfer learning` model, [5, 6, 7, 8, 9, 10], did better than 5% model for most of the labels.

Training setup	Model	Total	p0	p1	p2	p3	p4	A0	A1	A2
	English-Catalan **baseline**	0.8	0.72	0.49	0.33	0.06	0.0	0.05	0.04	0.0
`Transfer learning`	English-Catalan [1, 3, 4]	0.82	0.8	0.63	0.51	0.38	0.11	0.08	0.08	0.01
	English-Catalan [1, 2, 3]	0.81	0.78	0.56	0.41	0.28	0.22	0.08	0.07	0.02
	English-Catalan [1, 2, 3, 4]	0.82	0.82	0.64	0.48	0.41	0.2	0.09	0.08	0.02
	English-Catalan [6, 7]	0.79	0.74	0.52	0.37	0.16	0.0	0.06	0.05	0.0
	English-Catalan [5, 6, 7, 8, 9, 10]	0.77	0.66	0.43	0.25	0.03	0.0	0.04	0.02	0.0
	Catalan **baseline**	0.93	0.97	0.94	0.87	0.78	0.63	0.23	0.19	0.14
`Monolingual`	Catalan 50%	0.92	0.97	0.92	0.86	0.61	0.07	0.18	0.14	0.09
	Catalan 5%	0.86	0.91	0.55	0.21	0.0	0.0	0.02	0.02	0.01

Table 4.3: F1-scores for the models trained on Catalan data with the two training setups `Transfer learning` and `Monolingual`.
`Transfer learning`: Trained on English data, evaluated on Catalan data. **baseline**: No frozen parameters. For the other models in `Transfer learning` the number in [] indicate the layers that were frozen during training.
`Monolingual`: Trained on Catalan data, evaluated on Catalan data, with no frozen parameters. Model **baseline**: trained on the full data amount. The other two `Monolingual` models trained on 50% or 5% of the data amount.
All values are the average of five models' result.

Training setup	Total	p0	p1	p2	p3	p4	A0	A1	A2
English-Catalan [0]	0.8	0.72	0.49	0.33	0.06	0.0	0.05	0.04	0.0
English-Catalan [1]	0.81	0.78	0.6	0.43	0.26	0.04	0.07	0.07	0.01
English-Catalan [2]	0.81	0.77	0.61	0.46	0.34	0.0	0.08	0.07	0.01
English-Catalan [3]	0.8	0.75	0.57	0.43	0.31	0.05	0.08	0.07	0.01
English-Catalan [4]	0.8	0.76	0.56	0.44	0.28	0.01	0.07	0.06	0.01
English-Catalan [5]	0.8	0.73	0.54	0.36	0.17	0.01	0.06	0.06	0.0
English-Catalan [6]	0.8	0.76	0.55	0.41	0.25	0.01	0.07	0.06	0.0
English-Catalan [7]	0.79	0.74	0.52	0.38	0.18	0.0	0.06	0.05	0.0
English-Catalan [8]	0.8	0.74	0.53	0.41	0.15	0.0	0.06	0.05	0.0
English-Catalan [9]	0.8	0.74	0.53	0.4	0.19	0.0	0.06	0.05	0.0
English-Catalan [10]	0.8	0.75	0.54	0.43	0.21	0.0	0.07	0.05	0.01
English-Catalan [11]	0.8	0.75	0.55	0.43	0.26	0.0	0.07	0.06	0.01
English-Catalan [12]	0.8	0.73	0.5	0.34	0.09	0.0	0.05	0.04	0.0

Table 4.4: F1-scores for the models trained on Catalan data with individually frozen layers. The numbers in [] indicate which layers for the `Transfer learning` models that were frozen. [0] = No frozen layers. All values are the average of five models' results.

Table 4.2 with the results of models with single frozen layers too show a similar pattern to the Spanish models. Models with either layer 1, 2, 3 or 4 frozen are generally the best at predicting the correct label. The difference to the other layers is small however, especially for the argument labels.

4.3 German

The models in the German `Monolingual` setting performed on a low level compared to the models trained on data in the other languages. Notably, it was only for the labels p0 and A0 that the `Monolingual` models managed to achieve a F_1-score higher than 0.0. Lowering the amount of data did not have a substantial impact either as the results for the `Monolingual` **baseline** and the `Monolingual` model trained with 50% of the data are almost identical. The results of the `Transfer learning` setting differ as well from

Training setup	Model	Total	p0	p1	p2	p3	p4	A0	A1	A2
	English-German **baseline**	0.87	0.25	0.08	0.03	0.01	0.08	0.04	0.03	0.0
`Transfer learning`	English-German [1, 3, 4]	0.85	0.26	0.09	0.03	0.02	0.0	0.07	0.03	0.0
	English-German [1, 2, 3]	0.82	0.25	0.08	0.01	0.1	0.0	0.06	0.03	0.01
	English-German [1, 2, 3, 4]	0.85	0.25	0.08	0.02	0.01	0.0	0.06	0.03	0.0
	English-German [6, 7]	0.87	0.21	0.07	0.04	0.01	0.1	0.06	0.03	0.0
	English-German [5, 6, 7, 8, 9, 10]	0.84	0.27	0.1	0.07	0.08	0.0	0.04	0.03	0.0
`Monolingual`	German **baseline**	0.95	0.74	0.5	0.0	0.0	0.0	0.13	0.05	0.0
	German 50%	0.95	0.75	0.51	0.0	0.0	0.0	0.13	0.06	0.0
	German 5%	0.93	0.61	0.0	0.0	0.0	0.0	0.01	0.0	0.0

Table 4.5: F_1-scores for the models trained on German data with the two training setups `Transfer learning` and `Monolingual`.
`Transfer learning`: Trained on English data, evaluated on German data. **baseline**: No frozen parameters. For the other models in `Transfer learning` the number in [] indicate the layers that were frozen during training.
`Monolingual`: Trained on German data, evaluated on German data, with no frozen parameters. Model **baseline**: trained on the full data amount. The other two `Monolingual` models trained on 50% or 5% of the data amount.
All values are the average of five models' result.

Training setup	Total	p0	p1	p2	p3	p4	A0	A1	A2
English-German [0]	0.87	0.25	0.08	0.03	0.01	0.08	0.04	0.03	0.0
English-German [1]	0.86	0.26	0.08	0.03	0.02	0.03	0.06	0.03	0.01
English-German [2]	0.87	0.24	0.06	0.05	0.01	0.0	0.05	0.02	0.0
English-German [3]	0.87	0.26	0.1	0.06	0.01	0.0	0.04	0.03	0.0
English-German [4]	0.87	0.18	0.07	0.02	0.01	0.0	0.04	0.02	0.0
English-German [5]	0.87	0.21	0.07	0.02	0.0	0.0	0.03	0.02	0.0
English-German [6]	0.84	0.24	0.07	0.03	0.03	0.0	0.06	0.04	0.01
English-German [7]	0.86	0.26	0.07	0.04	0.01	0.04	0.05	0.04	0.0
English-German [8]	0.87	0.2	0.05	0.01	0.01	0.0	0.03	0.02	0.0
English-German [9]	0.89	0.22	0.06	0.06	0.0	0.0	0.03	0.02	0.0
English-German [10]	0.85	0.3	0.11	0.06	0.02	0.04	0.08	0.05	0.01
English-German [11]	0.87	0.25	0.07	0.07	0.01	0.07	0.04	0.04	0.0
English-German [12]	0.86	0.29	0.09	0.05	0.04	0.0	0.06	0.06	0.0

Table 4.6: F1-scores for the models trained on German data with individually frozen layers. The numbers in [] indicate which layers for the `Transfer learning` models that were frozen. [0] = No frozen layers. All values are the average of five models' results.

Training setup	Model	Total	p0	p1	p2	p3	p4	A0	A1	A2
Transfer learning	English-Chinese **baseline**	0.67	0.43	0.17	0.1	0.06	0.01	0.01	0.0	0.0
	English-Chinese [1, 3, 4]	0.67	0.47	0.2	0.12	0.13	0.06	0.01	0.0	0.0
	English-Chinese [1, 2, 3]	0.65	0.47	0.16	0.16	0.14	0.06	0.01	0.0	0.0
	English-Chinese [1, 2, 3, 4]	0.66	0.45	0.16	0.11	0.11	0.07	0.02	0.0	0.0
	English-Chinese [6, 7]	0.66	0.46	0.21	0.15	0.1	0.02	0.01	0.0	0.0
	English-Chinese [5, 6, 7, 8, 9, 10]	0.66	0.41	0.14	0.08	0.01	0.0	0.01	0.0	0.0
Monolingual	Chinese **baseline**	0.87	0.71	0.53	0.25	0.14	0.05	0.03	0.0	
	Chinese 50%	0.87	0.7	0.5	0.27	0.09	0.04	0.03	0.0	
	Chinese 5%	0.79	0.46	0.18	0.04	0.07	0.03	0.01	0.0	

Table 4.7: F1-scores for the models trained on Chinese data with the two training setups `Transfer learning` and `Monolingual`.
`Transfer learning`: Trained on English data, evaluated on Chinese data. **baseline**: No frozen parameters. For the other models in `Transfer learning` the number in [] indicate the layers that were frozen during training.
`Monolingual`: Trained on Chinese data, evaluated on Chinese data, with no frozen parameters. Model **baseline**: trained on the full data amount. The other two `Monolingual` models trained on 50% or 5% of the data amount.
All values are the average of five models' result.

the other languages' result. The German models trained with layers [5, 6, 7, 8, 9, 10] frozen outperformed the other `Transfer learning` models for most labels. Notably, neither layer combination performed extraordinarily good or bad, in fact all layer combinations achieved F_1-scores relatively close to each other. Even the model with zero layers frozen did well compared to the other models in the `Transfer learning` setting. As with Catalan, the `Transfer learning` models achieved higher F_1-scores than the low-resource `Monolingual` model.

As can be seen in Table 4.3, the F_1-scores for models with individually frozen layers mimic those of the models in the `Transfer learning` setting. There is not any layer that consistently makes the model perform better when frozen across all labels, with the possible exception of layer 10, which does outperform the other layers for labels p0, p1 and A0. It does quite poorly however for label p4.

4.4 Chinese

The models in the Chinese `Monolingual` setup got results more similar to the Spanish and Catalan models. The same pattern can be seen here where halving the data does not harm the result in a major way. Using only 5% of the data however causes severe reduction in F_1-scores for all labels except p0. The `Transfer learning` results have no specific frozen layer combination that stands out from the other models. [1, 3, 4] has the highest F_1-score for label p0 and p3, however the score for p0 is only 0.01 above the second highest score for that label and only 0.02 higher than the second highest score for p3. Likewise [1, 2, 3, 4] beats the second highest score for p4 and A0 by 0.01, and [6, 7] gets the highest F_1-score for p1 and p2 by 0.01 and 0.03 respectively. The models with the highest F_1-scores in the `Transfer learning` setting manages to beat the score of the `Monolingual` model with 5% of the data for label p3 and is tied with the score for label p4. All `Transfer learning` models except [5, 6, 7, 8, 9, 10] beat the baseline model with zero frozen layers.

With individual layers frozen, models trained on Chinese data do especially well when layer 2 is frozen and especially poorly when layer 12 is frozen. As can be seen in Table 4.4, the `Transfer learning` model with layers [6, 7] frozen did relatively well compared to the results of the languages. This is reflected when those layers are

Training setup	Total	p0	p1	p2	p3	p4	A0	A1	A2
English-Chinese [0]	0.67	0.43	0.17	0.1	0.06	0.01	0.01	0.0	0.0
English-Chinese [1]	0.66	0.45	0.17	0.13	0.08	0.03	0.01	0.0	0.0
English-Chinese [2]	0.66	0.46	0.21	0.14	0.11	0.04	0.02	0.0	0.0
English-Chinese [3]	0.67	0.45	0.16	0.12	0.09	0.02	0.01	0.0	0.0
English-Chinese [4]	0.67	0.45	0.17	0.14	0.1	0.02	0.01	0.0	0.0
English-Chinese [5]	0.67	0.43	0.2	0.13	0.08	0.02	0.01	0.0	0.0
English-Chinese [6]	0.67	0.43	0.16	0.12	0.11	0.04	0.01	0.0	0.0
English-Chinese [7]	0.67	0.43	0.15	0.12	0.06	0.01	0.02	0.0	0.0
English-Chinese [8]	0.67	0.45	0.18	0.13	0.09	0.04	0.02	0.0	0.0
English-Chinese [9]	0.67	0.41	0.13	0.08	0.05	0.01	0.01	0.0	0.0
English-Chinese [10]	0.67	0.43	0.14	0.08	0.07	0.02	0.01	0.0	0.0
English-Chinese [11]	0.67	0.44	0.15	0.11	0.06	0.01	0.02	0.0	0.0
English-Chinese [12]	0.67	0.42	0.16	0.09	0.07	0.01	0.01	0.0	0.0

Table 4.8: F1-scores for the models trained on Chinese data with individually frozen layers. The numbers in [] indicate which layers for the `Transfer learning` models that were frozen. [0] = No frozen layers. All values are the average of five models' results.

individually frozen as well, although their high performance is limited more to the higher order labels.

5 Discussion

Four conclusions can be drawn from the results presented in Chapter 4:

1) **When doing semantic role labeling through transfer learning, it is better to freeze layers of the pre-trained model than to not freeze them**, and for certain languages this method can even outperform models trained on lower amounts of data in the same language they are evaluated in. In Spanish for instance, the `Transfer learning` model trained with no frozen parameters only performed better than the model with layers 5-10 frozen. The other models outperformed the baseline model with no frozen layers by a wide margin, especially the predicates were much more accurately annotated by the models with frozen layers. The same pattern can be found as well in Catalan, where the best performing models with frozen layers even managed to beat the low-resource `Monolingual` setting. The `Transfer learning` Chinese and German models struggled compared to the Spanish and Catalan ones however, and performed much worse. Still, it proved better to freeze layers for SRL through transfer learning for these languages as well, as models with frozen layers achieved higher F_1-scores than the baselines with no frozen layers.

2) **Some layers are better to freeze than other layers**. For Spanish and Catalan (tables 4.1 and 4.2), the results clearly show that freezing the parameters in either of the first four layers makes the model perform much better for this task than is the case when the parameters in either of the middle or later layers are frozen. In fact, the F_1-scores across the labels show a rather steady decrease from layer 5 and below. Layer 6 and 11 break the pattern slightly with a small increase in primarily label p3. For Chinese (Table 4.4) the layers that prove well to freeze are more spread out than in the Catalan and Spanish models: Layer 2 performs at the top for all labels while layer 9 is consistently at the bottom for all labels. Apart from these, no clear locational pattern emerges however. The German data (Table 4.3) shows even fewer patterns for where the layers that make the model better or worse when frozen are located. For labels p0, p1 and A0, a frozen layer 10 is best, but for other labels other layers work better, with most layers not achieving a substantially higher F_1-score than others. Either way, it is still clear that some layers are better to freeze than other layers as layer 10 outperforms for instance layer 1 for all labels.

3) **It is generally better to freeze combinations of layers instead of just freezing one layer at a time**. Looking at the Spanish results, the models in the `Transfer learning` setting with some combination of the first four layers frozen performs on par with, or better than, models where either of the first four layers are frozen individually, depending on the label. For the first four predicate labels (p0, p1, p2, p3), the F_1-score of the layer combinations are rather similar to the F_1-scores of the models with individually frozen layers. Label p4 is however much more accurately labeled when multiple layers are frozen at the same time. A small increase can also be seen in the F_1-score of the argument labels. In the Catalan data, models that are trained with combinations of the first four layers frozen are much better at predicting predicate labels than model where the same layers are frozen one at a time. For the argument labels the difference is much smaller, but the results are generally also much lower here, so it is more difficult to properly analyze these results. The same can be said about the Chinese models. Predicates are generally more likely to be labeled correctly by the Chinese models if multiple layers are frozen during training rather if they are

frozen individually, but the scores for the arguments are very low, indicating that neither model type managed to learn the dataset enough to be able to consistently make accurate predictions. For the German data this is the case for all labels except label po and perhaps p1.

4) **Layers that improve SRL performance when frozen are not layers with high attention weights for SRL tasks**. The heatmaps in figures 5.1 to 5.4 show the attention weights of predicate-argument tokens in each attention head in mBERT, averaged for each sentence in the dataset. The average attention weight per layer for SRL, i.e. the sum of the average weights of all heads in a layer, varies slightly between the languages. For SRL on the German data, the weights are more evenly spread out across the model compared to how it is for the other languages. For instance, the difference between the layer with the highest weights and the layer with the lowest weights is 0.64 for the German data and between 1.03 and 1.3 for the other three languages. Either way, the distribution of the weights is very similar regardless of language, with the middle layers containing the highest attention weights and the lowest weight being located in the outer layers. In the first four layers the attention weights tend to be especially low, and the weights in layer 2 are the lowest for all four languages.

As the tables in Section 4 show, the layers that increase the performance when frozen are for Spanish, Catalan and Chinese layers 1, 2, 3 and 4. For the same languages, the layers that decrease the performance when frozen are layers 5, 6, 7, 8, 9 and 10. For German, admittedly, the situation is the opposite: the first four layers when frozen either decrease the F_1-score or at least make no improvements, and the middle layers, i.e. layers 5-10, cause an increase in F_1-score compared to the baseline with no frozen layers. It is therefore rather clear that the hypothesis stated in Section 1 does not hold: **layers with high attention weights for an SRL task do not improve performance if frozen** (or at least not for all languages).

The relationship between the attention weights of a layer and whether a layer should have its parameters updated during training might instead be the opposite from the original hypothesis. The lower the attention weight of a layer in a pre-trained model, the more likely it is that the model will benefit from freezing the parameters of that layer. What speaks in favour of this hypothesis is the following:

1) Regardless of language, the attention weights of the heads in layer 2 are the lowest.

2) For Chinese (Table 4.4) and Catalan (Table 4.2), the best performing models had the parameters of layer 2 frozen. For the models trained on Spanish data (Table 4.1) the model with layer 2 frozen performed almost as well as the best performing model.

Looking only at layer 2, there therefore appears to be support for the hypothesis that freezing the parameters of a layer where the attention weight is low will lead to an increase of F_1-score.

Against this hypothesis is the fact that if a lower layer attention weight equaled a higher F_1-score, then the opposite should also be true: the layers where the attention weight is high should lead to a decrease in the model's F_1-score when frozen. This does not seem to be case, as for all languages layers 6 and 7 have the highest attention weights, but for no language do the model with layers 6 and 7 frozen have the lowest F_1-score. It therefore appears that it is not the SRL attention weights that determines which layers' parameters that work best to freeze during zero-shot SRL.

Instead it could be that the reason to why the first four layers provide the highest performance increase when frozen is due to the fact the upper layers in a BERT model tends to be more focused on surface level tasks (Jawahar et al., 2019), and that for

a more complex linguistic area such as semantics, these are not needed to the same extent. Not updating the parameters of these surface layers during training enables the model to instead optimize later layers that are more essential to the task. This would also mean that individual layers that decrease the score when frozen, such as layer 9 for Chinese data, is more important than the other layers for semantic role labeling.

Several studies investigating individual layers and attention heads in BERT found that for semantic tasks the entire model is used to perform the task, and that no single layer is especially necessary for the model's ability to perform well on semantics (Clark et al., 2019; de Vries et al., 2020; Jawahar et al., 2019; Tenney, Das, et al., 2019). Our results contradict these claims to an extent given that the first four layers seem at least to not matter as much for SRL.

Using heatmaps of attention weights to determine the functioning of the model can be seen as a form of explanation in AI, where the weights are used as a way to determine how the model operates internally. If it were possible to use the heatmaps as a form of explanation, there should have been some form of correlation between high/low attention weights and higher/lower F_1-score when freezing a layer. But as explained above this is not the case. As such, these experiments give support to such positions as Jain and Wallace (2019), that it is not possible to use attention as a means of explanation in AI.

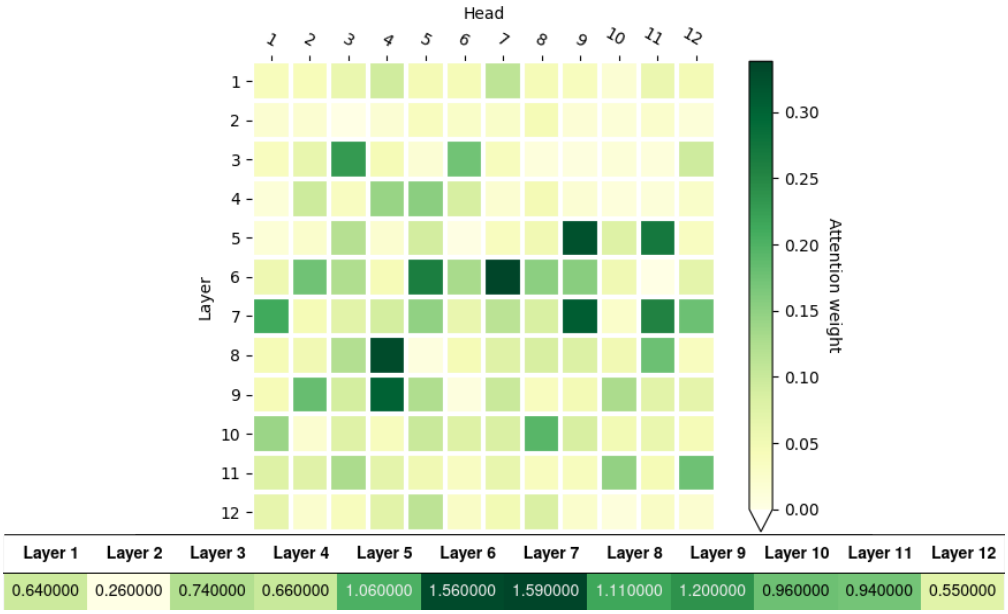

Figure 5.1: Heatmap of the average attention weights for the Spanish CoNLL-2009 data. Bottom of image shows the average weights for each layer's attention heads.

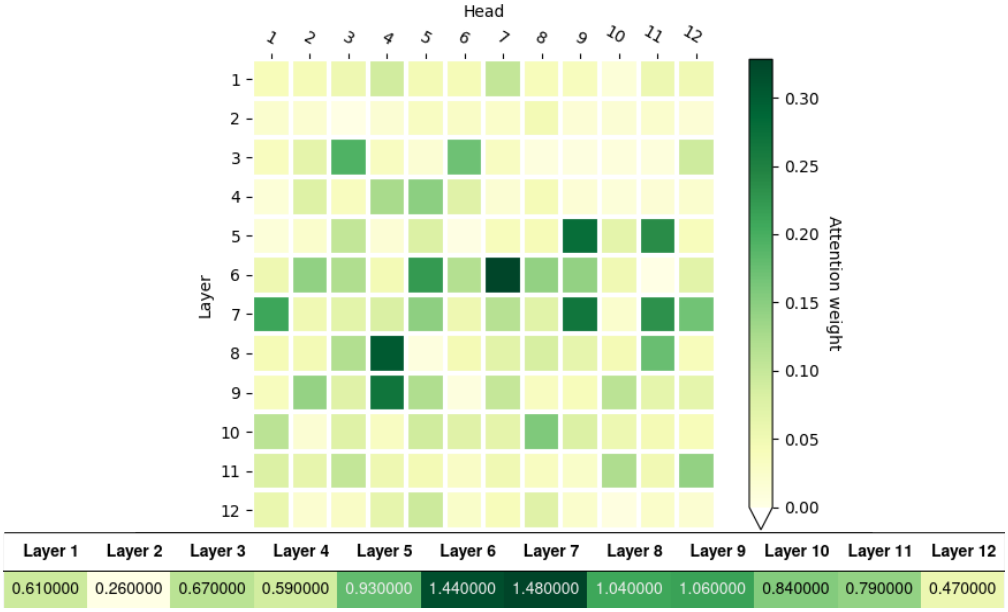

Figure 5.2: Heatmap of the average attention weights for the Catalan CoNLL-2009 data. Bottom of image shows the average weights for each layer's attention heads.

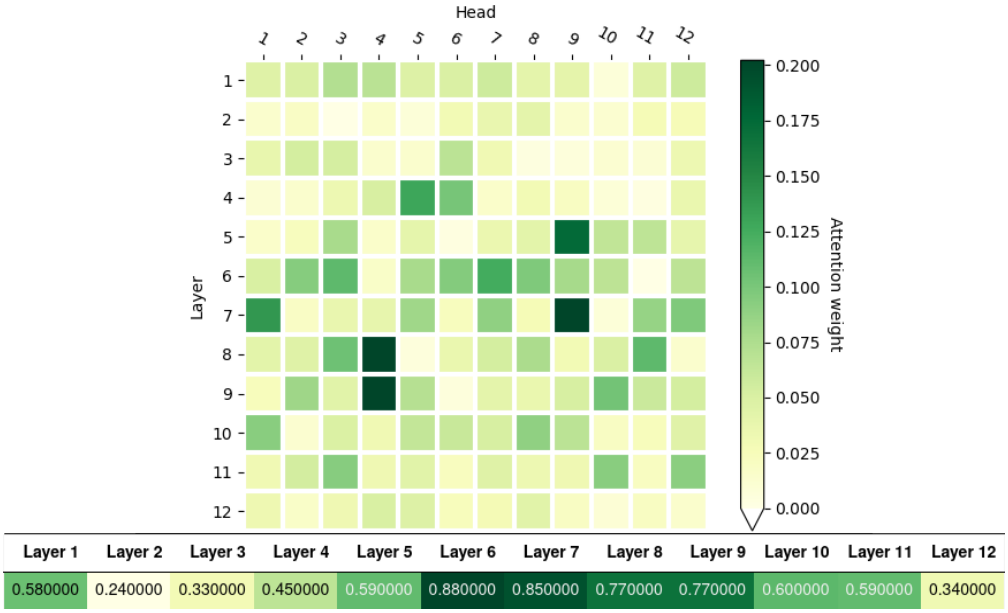

Figure 5.3: Heatmap of the average attention weights for the German CoNLL-2009 data. Bottom of image shows the average weights for each layer's attention heads.

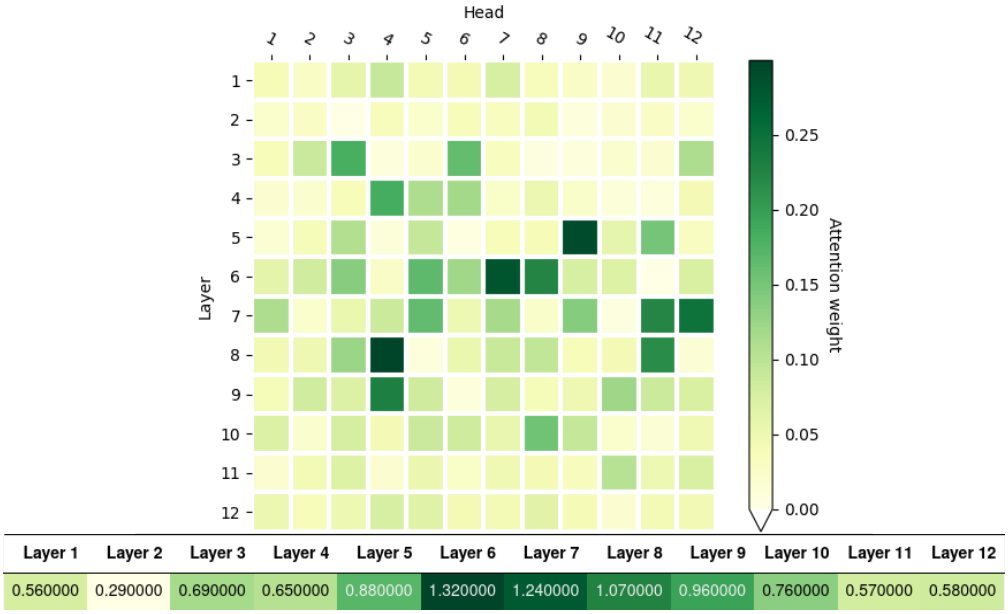

Figure 5.4: Heatmap of the average attention weights for the Chinese CoNLL-2009 data. Bottom of image shows the average weights for each layer's attention heads.

6 Conclusions and future work

This study investigated the possibilities of utilizing the transfer-learning capabilities of large language models in order to overcome the lack of annotated SRL-datasets for low-resource languages. In addition we also experimented with freezing parameters in different layers of the model to see if this could improve the transfer. The results concluded that while models where surprisingly robust even when only trained on half of the data size, transfer-learning SRL could in certain cases outperform truly low-resource baseline. For Catalan, the transfer-learning model trained on English and evaluated on Catalan, without updating the parameters in the first four layer, outperformed the model trained on 5 % of the Catalan data. We hypothesized that the layers that would be beneficial to freeze would be layers with high average attention weights for an SRL task. That is, the layers that where already well adapted to the task and did not need further training. Comparisons with heatmaps of average attention weights of the models' attention heads showed that it instead was the case that it was rather layers whose function was not needed for the task that proved beneficial to freeze. The first four layers in BERT are focused on surface level task, which are not as necessary to more complex linguistic areas such as semantics. As such, the models did benefit from not having to update the parameters on those layers and instead focus on the layers that were of more importance to the task.

Areas of future work for this task are many. This thesis does not go into detail on the linguistic reasons for the different results of the languages. Spanish and Catalan, two very similar languages performed better than the more different languages German and Chinese. Further studies could investigate the linguistic reasons to why these languages did not perform on the same level as Spanish and Catalan. Also, this study uses a rather outdated pre-trained model. While more difficult, it would be an interesting ares of research to see how well these results translate to a scaled up model with parameter sizes closer to the large language models of today.

Bibliography

Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio (2015). "Neural Machine Translation by Jointly Learning to Align and Translate". In: *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*. Ed. by Yoshua Bengio and Yann LeCun. URL: http://arxiv.org/abs/1409.0473.

Baker, Collin F., Charles J. Fillmore, and John B. Lowe (1998). *The Berkeley FrameNet Project*. Montreal, Quebec, Canada, Aug. 1998. DOI: 10.3115/980845.980860. URL: https://aclanthology.org/P98-1013.

Bruton, Micaella and Meriem Beloucif (2023). "BERTie Bott's Every Flavor Labels: A Tasty Introduction to Semantic Role Labeling for Galician". In: *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*. Ed. by Houda Bouamor, Juan Pino, and Kalika Bali. Singapore: Association for Computational Linguistics, Dec. 2023, pp. 10892–10902. DOI: 10.18653/v1/2023.emnlp-main.671. URL: https://aclanthology.org/2023.emnlp-main.671.

Chowdhury, Amartya, Deepak K. T., Samudra Vijaya K, and S. R. Mahadeva Prasanna (2022). "Machine Translation for a Very Low-Resource Language - Layer Freezing Approach on Transfer Learning". In: *Proceedings of the Fifth Workshop on Technologies for Machine Translation of Low-Resource Languages (LoResMT 2022)*. Ed. by Atul Kr. Ojha, Chao-Hong Liu, Ekaterina Vylomova, Jade Abbott, Jonathan Washington, Nathaniel Oco, Tommi A Pirinen, Valentin Malykh, Varvara Logacheva, and Xiaobing Zhao. Gyeongju, Republic of Korea: Association for Computational Linguistics, Oct. 2022, pp. 48–55. URL: https://aclanthology.org/2022.loresmt-1.7.

Clark, Kevin, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning (2019). "What Does BERT Look at? An Analysis of BERT's Attention". In: *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*. Ed. by Tal Linzen, Grzegorz Chrupała, Yonatan Belinkov, and Dieuwke Hupkes. Florence, Italy: Association for Computational Linguistics, Aug. 2019, pp. 276–286. DOI: 10.18653/v1/W19-4828. URL: https://aclanthology.org/W19-4828.

Conneau, Alexis, German Kruszewski, Guillaume Lample, Loïc Barrault, and Marco Baroni (2018). "What you can cram into a single $&!#* vector: Probing sentence embeddings for linguistic properties". In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Ed. by Iryna Gurevych and Yusuke Miyao. Melbourne, Australia: Association for Computational Linguistics, July 2018, pp. 2126–2136. DOI: 10.18653/v1/P18-1198. URL: https://aclanthology.org/P18-1198.

Deshpande, Ameet and Karthik Narasimhan (2020). "Guiding Attention for Self-Supervised Learning with Transformers". In: *Findings of the Association for Computational Linguistics: EMNLP 2020*. Ed. by Trevor Cohn, Yulan He, and Yang Liu. Online: Association for Computational Linguistics, Nov. 2020, pp. 4676–4686. DOI: 10.18653/v1/2020.findings-emnlp.419. URL: https://aclanthology.org/2020.findings-emnlp.419.

Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova (2019). "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and*

Short Papers). Ed. by Jill Burstein, Christy Doran, and Thamar Solorio. Minneapolis, Minnesota: Association for Computational Linguistics, June 2019, pp. 4171–4186. DOI: 10.18653/v1/N19-1423. URL: https://aclanthology.org/N19-1423.

de Vries, Wietse, Andreas van Cranenburgh, and Malvina Nissim (2020). "What's so special about BERT's layers? A closer look at the NLP pipeline in monolingual and multilingual models". In: *Findings of the Association for Computational Linguistics: EMNLP 2020*. Ed. by Trevor Cohn, Yulan He, and Yang Liu. Online: Association for Computational Linguistics, Nov. 2020, pp. 4339–4350. DOI: 10.18653/v1/2020.findings-emnlp.389. URL: https://aclanthology.org/2020.findings-emnlp.389.

Ding, Yanzhuo, Yang Liu, Huanbo Luan, and Maosong Sun (2017). "Visualizing and Understanding Neural Machine Translation". In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Ed. by Regina Barzilay and Min-Yen Kan. Vancouver, Canada: Association for Computational Linguistics, July 2017, pp. 1150–1159. DOI: 10.18653/v1/P17-1106. URL: https://aclanthology.org/P17-1106.

Givón, Talmy (2001). *Syntax: An Introduction, vol. I*. new edition of Syntax: A functional-typological introduction, 1984. Amsterdam; Philadelphia: John Benjamins.

Ha, T.-L. N. My-Linh, V.-H. Nguyen, T.-M.-H. Nguyen, P. Le-Hong, and T.-H. Phan (2014). "Building a semantic role annotated corpus for Vietnamese". In: *Proceedings of the 17th National Symposium on Information and Communication Technology*.

Hajič, Jan, Massimiliano Ciaramita, Richard Johansson, Daisuke Kawahara, Maria Antònia Martí, Lluís Màrquez, Adam Meyers, Joakim Nivre, Sebastian Padó, Jan Štěpánek, Pavel Straňák, Mihai Surdeanu, Nianwen Xue, and Yi Zhang (2009). "The CoNLL-2009 Shared Task: Syntactic and Semantic Dependencies in Multiple Languages". In: *Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL 2009): Shared Task*. Ed. by Jan Hajič. Boulder, Colorado: Association for Computational Linguistics, June 2009, pp. 1–18. URL: https://aclanthology.org/W09-1201.

Hassid, Michael, Hao Peng, Daniel Rotem, Jungo Kasai, Ivan Montero, Noah A. Smith, and Roy Schwartz (2022). "How Much Does Attention Actually Attend? Questioning the Importance of Attention in Pretrained Transformers". In: *Findings of the Association for Computational Linguistics: EMNLP 2022*. Ed. by Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, Dec. 2022, pp. 1403–1416. DOI: 10.18653/v1/2022.findings-emnlp.101. URL: https://aclanthology.org/2022.findings-emnlp.101.

He, Luheng, Kenton Lee, Mike Lewis, and Luke Zettlemoyer (2017). "Deep semantic role labeling: What works and what's next". In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 473–483.

Jain, Sarthak and Byron C. Wallace (2019). "Attention is not Explanation". In: *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)*. Ed. by Jill Burstein, Christy Doran, and Thamar Solorio. Minneapolis, Minnesota: Association for Computational Linguistics, June 2019, pp. 3543–3556. DOI: 10.18653/v1/N19-1357. URL: https://aclanthology.org/N19-1357.

Jawahar, Ganesh, Benoıt Sagot, and Djamé Seddah (2019). "What Does BERT Learn about the Structure of Language?" In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Ed. by Anna Korhonen, David Traum, and Lluís Màrquez. Florence, Italy: Association for Computational Linguistics, July 2019, pp. 3651–3657. DOI: 10.18653/v1/P19-1356. URL: https://aclanthology.org/P19-1356.

Jurafsky, Daniel and James H. Martin (2023). *Speech and Language Processing*. 3rd. Draft. Stanford.

Kaplan, Jared, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei (2020). *Scaling Laws for Neural Language Models*. arXiv: 2001.08361 [cs.LG].

Kovaleva, Olga, Alexey Romanov, Anna Rogers, and Anna Rumshisky (2019). "Revealing the Dark Secrets of BERT". In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Ed. by Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan. Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 4365–4374. DOI: 10.18653/v1/D19-1445. URL: https://aclanthology.org/D19-1445.

Levin, Beth (1993). *English Verb Classes and Alternations: A Preliminary Investigation*. University of Chicago Press.

Michel, Paul, Omer Levy, and Graham Neubig (2019). "Are Sixteen Heads Really Better than One?" In: *Advances in Neural Information Processing Systems*. Ed. by H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett. Vol. 32. Curran Associates, Inc. URL: https://proceedings.neurips.cc/paper_files/paper/2019/file/2c601ad9d2ff9bc8b282670cdd54f69f-Paper.pdf.

Mikolov, Tomas, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean (2013). "Distributed Representations of Words and Phrases and their Compositionality". In: *Advances in Neural Information Processing Systems*. Ed. by C.J. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K.Q. Weinberger. Vol. 26. Curran Associates, Inc. URL: https://proceedings.neurips.cc/paper_files/paper/2013/file/9aa42b31882ec039965f3c4923ce901b-Paper.pdf.

Palmer, Martha, Daniel Gildea, and Paul Kingsbury (2004). "PropBank: The Next Level of TreeBank". In: *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC-2004)*. European Language Resources Association (ELRA), pp. 1395–1398.

Pennington, Jeffrey, Richard Socher, and Christopher D. Manning (2014). "GloVe: Global Vectors for Word Representation". In: *Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1532–1543. URL: http://www.aclweb.org/anthology/D14-1162.

Peters, Matthew E., Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer (2018). "Deep Contextualized Word Representations". In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. Ed. by Marilyn Walker, Heng Ji, and Amanda Stent. New Orleans, Louisiana: Association for Computational Linguistics, June 2018, pp. 2227–2237. DOI: 10.18653/v1/N18-1202. URL: https://aclanthology.org/N18-1202.

Radford, Alec, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever (2018). "Improving language understanding by generative pre-training".

Raffel, Colin, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu (2020). "Exploring the limits of transfer learning with a unified text-to-text transformer". *J. Mach. Learn. Res.* 21.1 (Jan. 2020). ISSN: 1532-4435.

Rogers, Anna, Olga Kovaleva, and Anna Rumshisky (2020). *A Primer in BERTology: What we know about how BERT works*. arXiv: 2002.12327 [cs.CL].

Shi, Peng and Jimmy Lin (2019). *Simple BERT Models for Relation Extraction and Semantic Role Labeling*. arXiv: 1904.05255 [cs.CL].

Tan, Zhixing, Mingxuan Wang, Jun Xie, Yidong Chen, and Xiaodong Shi (2018). "Deep semantic role labeling with self-attention". In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 32. 1.

Tenney, Ian, Dipanjan Das, and Ellie Pavlick (2019). "BERT Rediscovers the Classical NLP Pipeline". In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Ed. by Anna Korhonen, David Traum, and Lluís Màrquez. Florence, Italy: Association for Computational Linguistics, July 2019, pp. 4593–4601. DOI: 10.18653/v1/P19-1452. URL: https://aclanthology.org/P19-1452.

Tenney, Ian, Patrick Xia, Berlin Chen, Alex Wang, Adam Poliak, R Thomas McCoy, Najoung Kim, Benjamin Van Durme, Samuel R. Bowman, Dipanjan Das, and Ellie Pavlick (2019). *What do you learn from context? Probing for sentence structure in contextualized word representations*. arXiv: 1905.06316 [cs.CL].

Tjong Kim Sang, Erik F. (2002). "Introduction to the CoNLL-2002 Shared Task: Language-Independent Named Entity Recognition". In: *COLING-02: The 6th Conference on Natural Language Learning 2002 (CoNLL-2002)*. URL: https://aclanthology.org/W02-2024.

Touvron, Hugo, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample (2023). *LLaMA: Open and Efficient Foundation Language Models*. arXiv: 2302.13971 [cs.CL].

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin (2017). "Attention is All you Need". In: *Advances in Neural Information Processing Systems*. Ed. by I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett. Vol. 30. Curran Associates, Inc.

Voita, Elena, David Talbot, Fedor Moiseev, Rico Sennrich, and Ivan Titov (2019). "Analyzing Multi-Head Self-Attention: Specialized Heads Do the Heavy Lifting, the Rest Can Be Pruned". In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Ed. by Anna Korhonen, David Traum, and Lluís Màrquez. Florence, Italy: Association for Computational Linguistics, July 2019, pp. 5797–5808. DOI: 10.18653/v1/P19-1580. URL: https://aclanthology.org/P19-1580.

Wang, Shanshan, Zhumin Chen, Zhaochun Ren, Huasheng Liang, Qiang Yan, and Pengjie Ren (2022). *Paying More Attention to Self-attention: Improving Pre-trained Language Models via Attention Guiding*. arXiv: 2204.02922 [cs.CL].

Williams, Adina, Nikita Nangia, and Samuel Bowman (2018). "A Broad-Coverage Challenge Corpus for Sentence Understanding through Inference". In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. Ed. by Marilyn Walker, Heng Ji, and Amanda Stent. New Orleans, Louisiana: Association for Computational Linguistics, June 2018, pp. 1112–1122. DOI: 10.18653/v1/N18-1101. URL: https://aclanthology.org/N18-1101.

Zhou, Jie and Wei Xu (2015). "End-to-end learning of semantic role labeling using recurrent neural networks". In: *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 1127–1137.

Zhu, Yukun, Ryan Kiros, Richard Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler (2015). *Aligning Books and Movies: Towards Story-like Visual Explanations by Watching Movies and Reading Books*. arXiv: 1506.06724 [cs.CV].

Zoph, Barret, Deniz Yuret, Jonathan May, and Kevin Knight (2016). "Transfer Learning for Low-Resource Neural Machine Translation". In: *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Ed. by Jian Su, Kevin Duh, and Xavier Carreras. Austin, Texas: Association for Computational Linguistics, Nov. 2016, pp. 1568–1575. DOI: 10.18653/v1/D16-1163. URL: https://aclanthology.org/D16-1163.