Universität Zürich[UZH]

Masterarbeit

zur Erlangung des akademischen Grades

Master of Arts

der Philosophischen Fakultät der Universität Zürich

[REL Enriching] [ARG1-GOL BERT Embeddings] [ARG2-PPT with Semantic Role Labels] [ARGM-GOL for Natural Language Understanding Tasks in German]

Verfasserin/Verfasser: [ARG0-PAG Jonathan Schaber]

Matrikel-Nr: 11-771-359

Referentin/Referent: Dr. Simon Clematide

[Betreuerin/Betreuer: (Titel Vorname Name) [nur falls vom Ref. unterschiedlich]]

Institut für Computerlinguistik

Abgabedatum: April 26, 2021

Abstract

This is the place to put the English version of the abstract.

Zusammenfassung

Und hier sollte die Zusammenfassung auf Deutsch erscheinen.

Acknowledgement

I want to thank Simon Clematide, Y and Z for their precious help. And many thanks to whoever for proofreading the present text.

Contents

List of Figures

List of Tables

List of Acronyms

BERT	Bidirectional Encoder Representations from Transformers
BLEU	Bilingual Evaluation Understudy
BPE	Byte Pair Encoding
BOW	Bag of Words
CPOSTAG	Coarse-grained Part-Of-Speech tag
CNN	Convolutional Neural Network
deISEAR	German International Survey on Emotion Antecedents and Reactions
FFNN	Fully-connected Feed Forward Neural Network
GIGO	Garbage In, Garbage Out
GRU	Gated Recurrent Unit
LCS	Lexical Conceptual Structure
LSTM	Long Short-Term Memory
ML	Machine Learning
NLP	Natural Language Processing
NLU	Natural Language Understanding
OOD	Out-of-Domain
POS	Part-Of-Speech / Poverty of Stimulus
POSTAG	Fine-grained Part-Of-Speech tag
RNN	Recurrent Neural Network
SCARE	Sentiment Corpus of App Reviews
SOTA	State of the Art
SRL	Semantic Role Labelling / Semantic Role Labeller
STTS	Stuttgart-Tübingen-TagSet
USD	Universal Stanford Dependencies

1 Introduction

In this chapter, I will expand a bit further to put the intent of this master's thesis into a bigger picture. Therefore, I will line out the general problems and topics of NLU and elaborate a little bit on the methods and techniques that were developed to address those questions.

Further, I will tie in my approach with the current research efforts in NLU.

1.1 Motivation

Human language bears some truly mesmerizing features and puzzles, a lot of them are still not yet understood in all its depths: For example, it is still unclear how children are able to learn the grammar of their mother tongue from the corrupted and comparatively scarce language material they are exposed to (cf. Lust [2006]; Lewis and Elman [2001]).[1]

But maybe the most trivial and enigmatic trait about human language is that we actually *understand* each other so well: That during a discourse, person X can retrieve the intentioned meaning of expressions uttered by person Y, and vice versa. Further, we are able to logically deduce a whole lot information that is not explicitly

[1] The famous term describing this phenomenon, "Poverty of Stimulus" (POS), was coined by Chomsky [1980], arguing for an innate language learning/processing faculty. This has led to a fierce debate over that issue: Many researchers claim that the POS-motivated necessity for such an innate, human-specific, genetic trait does not hold, and humans learn language simply by means of extremely sophisticated statistic analysis. It has indeed been shown that infants are amazingly apt at extracting statistical information from auditive input [Saffran et al., 1996]. However, as was countered by Chomsky and others, this does not fully explain the ability of children to apply these statistical input cues to *hierarchical* structures : "The issue that is so central to this particular POS problem is tacit knowledge by the child that grammatical rules apply to such [hierarchical] structures" [Berwick et al., 2013] — there is no behavioral explanation for the tacit knowledge about hierarchical structures in the mind of the child. Therefore, the POS argument still holds, although it has been somewhat weakend by empirical findings about acquisition driven by statisitcal insights.

stated in an expression, and uphold such a state of affairs during the whole conversation. In fact, communication is perceived as such an intrinsic human capability, that it is often used as a proxy for attesting intelligence to an entity: Turing [1950] suggested his renowned "Turing-Test" as method for determining the capabilities of artificial intelligence systems: If the communicative behavior of an agent is indistinguishable from a human agent, it is justifiable, according to this view, to attribute "intelligence" to such a system.

That reproducing human communication is not as trivial as it might look like on first glance, reveal the following considerations, which are only a subset of all the complexities involved in human communication:

Vagueness We have no problem dealing with vague statements like "Most people never heard of it"; in a specific conversation situation we rely on extra-linguistic cues like pragmatics and common world knowledge to decide if "most" means 7.5 billion people, or 70% of our friends, or if the approximate number is even relevant (maybe it was ironically spoken, etc.)

Ambiguity The fact that a linguistic sign can't be interpreted in only one way, is an ubiquitous phenomenon in human language that is present on all levels: phonological, lexical, syntactic, semantic, pragmatic. A classic example for syntactic ambiguity would be the phrase "He saw the elephant with a telescope".

Corruption Contrary to how we perceive language when speaking to each other, utterances are mostly not well-formed, grammatical sentences, but show a varying degree of stutter, incomplete phrases, repetitions, and other "mistakes". While not as present in written language, depending on the domain, there still is quite some noise present, e.g. in online chat threads etc. Still, we mostly have no problems at all, reconstructing the encoded information from such a corrupted signal.

Common world knowledge Normally the information we encode in ordinary conversation is highly condensed and as scarce as possible — most of the actual information is reconstructed by the receiver, making use of general knowledge about the world (factual knowledge, such as "Bern is the capital of Switzerland", logical consequence such as "If X was at the party yesterday evening, then X was not in his apartment yesterday evening", etc.) and the actual situation, time, and place the conversation takes place.

So, every system that is built aiming at processing natural language in a "deep", human-like manner must cope with this inherent fuzzinesses of human communica-

tion. In the field of applied computational linguistics, often referred to as Natural Language Processing (NLP), this subfield of research is known as Natural Language Understanding (NLU).[2] NLU, therefore aims at producing systems which are able to retreive the semantic content encoded in natural language and are able to further act upon it: For example, a chatbot should be capable of "understanding" that the questions "What's the weather like?", "Can you tell me today's weather forecast, please?", "Will I need an umbrella today?" all have more or less the same meaning and should provoke the same answer.[3]

1.1.1 History, Methods, Problems of NLU

During the first phase of NLP, approximately from the 1950ies until the 1980ies, systems that addressed NLU problems were architectures that consisted of carefully hand-written symbolic grammars and knowledge bases that aimed at tackling a specific problem, such as recognizing textual entailment, coreference resolution, sentiment analysis, and so on.

From the 90ies on, the so-called emphstatistical revolution took place, and NLU related problems were now being addressed by learning patterns from huge data collections. One driver of this paradigm shift were the various difficulties the traditional systems bore: Their development "requiring a great deal of domain-specific knowledge engineering. In addition, the systems were brittle and could not function adequately outside the restricted tasks for which they were designed" [Brill and Mooney, 1997, p. 13]. The new, statistical approach tries to tackle NLU-problems by shifting the focus from tedious hand-crafting "to empirical, or corpus-based, methods

[2] Note that I will not engage in the discussion about whether it is philosophically appropriate to claim that computational models "understand" human language. A lot of controversy has arisen around this issue, with positions ranging from completely denying language models any sort of (linguistic) understanding [Bender et al., 2021] to the current trend of simply concentrating on beating NLU SOTAs with ever larger models and blindly taking this as proof of building architectures capable of learning human language. I would argue that the thruth lies somewhere in between: Of course it is not enough to perform well on a standardized data set to speak of "understanding", however, as Sahlgren and Carlsson [2021] point out: Also in humans, especially children, we measure language competences indirectly "by using various language proficiency tests, such as vocabulary tests, cloze tests, reading comprehension, as well as various forms of production, interaction, and mediation tests [...]".

[3] Although one could argue that the third question differs from the first two since it is a polar question; i.e. a simple "yes" or "no" would be grammatically correct — however, I have the strong feeling one would perceive this as a very dry, or even rude, answer and would expect a more elaborated answer in a regular conversation context.

in which development is much more data driven and is at least partially automated by using statistical or machine-learning methods to train systems on large amounts of real language data" [Brill and Mooney, 1997, p. 13]. The main challenge for engineers and scientists now laid in discovering suitable features, according to which the algorithm would hopefully learn helpful patterns from the language data for solving the task at hand. With this orientation towards data-driven NLP solutions came also the possibility to compare different architectures on the same standardized data set and measured

For half a decade now a next stage in NLU and NLP in general was entered: we are now in the middle of the *neural age* of computational linguistics: "Deep Learning waves have lapped at the shores of computational linguistics for several years now, but 2015 seems like the year when the full force of the tsunami hit the major Natural Language Processing (NLP) conferences." [Manning, 2015, p. 701]

In contrast to the statistical period's main challenge — the identification and extraction of suitable features —, now the algorithms are itself learning the features that are the most informative for a given task. The human part in the process is reduced to design the overall model architecture and compile large enough amounts of data that are, in the best case, also of good quality.

While the roughly sketched methods above apply to a wide ranges of applications in NLP, I will now point to some of the enquiries NLU aims at: Simply put, NLP is concernced with the structural side of natural language text, while NLU looks at the content of these utterances. For example, typical NLP tasks such as dependency parsing, POS-tagging, and coreference soltution don't require a semantic representation of words or phrases — often, it suffices to look at structural properties such as morphology, simple frequency statstics, or transition probablities to solve such problems to an acceptable extent. On the other hand, NLU tries to process language more in a manner as humans do it: We infer something from language, answer questions, detect logical inconsistencies etc.

A concise summary of the scope of the NLU and the numerous, non-intuitive pitfalls is given by McShane [2017] in her paper on NLU in cognitive agents, where she also argues that truly NLU-capable systems's abilities must go beyond "mere" symbol processing:

> cognitive agents must be nimble in the face of incomplete interpretations since even people do not perfectly understand every aspect of every utterance they hear. This means that once an agent has reached the best interpretation it can, it must determine how to proceed — be that act-

> ing upon the new information directly, remembering whatever it has understood and waiting to see what happens next, seeking out information to fill in the blanks, or asking its interlocutor for clarification. The reasoning needed to support NLU extends far beyond language itself, including, nonexhaustively, the agent's understanding of its own plans and goals; its dynamic modeling of its interlocutor's knowledge, plans, and goals, all guided by a theory of mind; its recognition of diverse aspects of human behavior, such as affect, cooperative behavior, and the effects of cognitive biases; and its integration of linguistic interpretations with its interpretations of other perceptive inputs, such as simulated vision and nonlinguistic audition. Considering all of these needs, it seems hardly possible that fundamental NLU will ever be achieved through the kinds of knowledge-lean text-string manipulation being pursued by the mainstream natural language processing (NLP) community. Instead, it requires a holistic approach to cognitive modeling of the type we are pursuing in a paradigm called OntoAgent.

All disagreements about the very nature of "understanding" and "communication" set aside, there is probably consensus that we are far from being able to construct a system that would pass the Turing test. As laid out in the quote above, such a model would need to be equipped with

1.1.2 Subtasks of Solving NLU

As the endeavor of building a holistic, fully NLU capable agent is probably an objective too ambitious to achieve at once, the strategy in modern computational linguistics is to break it down into smaller "subproblems":

Several core NLU skills have been identified and datasets tailored at those have been constructed.

For example:

Sentiment Detection Given a sentence, we can normally assess if the emotion transmitted by it is rather positive, negative, or neutral: "Oh my, what a lovely day!" is clearly positive, while "asdasd" probably is not.

Grammaticality Recognition Being able to understand utterances in a language also implies being able to judge the grammaticality of any statement in that language: "Eagles that fly can eat" is judged as valid by English speakers while "Eagles that fly eat can" is not.

Entailment Recognition An important ability of understanding is the recognition of the logical relationship in which two utterances stand. In other words, given two sentences A and B, does B follow from A? For example, given the sentence "The weather forecast predicts rainfall the next days" does the sentence "Tomorrow will be a beautiful, sunny day" conform, contradict or stand in a neutral relation to the first sentence.

Question Answering To locate a (or rather, the) relevant text span in a given context according to some questions is also a task where one would assume that some sort of understanding is needed: Given the question " What was the name of the King of England?" and the context "Henry V (16 September 1386 – 31 August 1422) was King of England from 1413 until his death in 1422.", a system would would need to extract the span "Henry V".

Datasets addressing those tasks are listed in table reftab:original-GLUE describing the GLUE benchmark dataset compilation.

1.1.3 Text Representations

Text can be analysed and represented in various ways.[4] For example, one could represent each word of a sentence as a number, e.g. the page number of the Oxford English Dictionary on which the definition of the word — or its lemma base form — is given: <Every> <event> <has> <a> <cause> → <234> <229> <388> <12> <176>. However, that would probably not be very informative, but if we represent each word of a sequence by its POS, we might be able to get some information out of it; especially if both representations, the "normal" one and the POS one, are combined: <Every DET> <event NOUN> <has VERB> <a DET> <cause NOUN>.

Embeddings

"Where has Deep Learning helped NLP? The gains so far have not so much been from true Deep Learning (use of a hierarchy of more abstract representations to promote generalization) as from the use of distributed word representations—through the use of real-valued vector representations of words and concepts. Having a dense, multi-dimensional representation of similarity between all words is incredibly useful in

[4]For simplicity, I will focus on written language in my thesis, however this applies to other modalities like speech and signing as well.

NLP, but not only in NLP." [Manning, 2015, p. 703]

Most NLU models don't operate on the text as it is, i.e. as an array of UTF-8 encoded signs. Often, it is easier to implement an algorithm that process text in some way, to encode this text numerically. A strategy adopted quite early is the so-called Bag of Words [Harris, 1954] technique, which encoded a sequence of words s in one vector, basically indicating what items of a given vocabulary or present in s. While this BOW technique is quite effective for certain tasks, e.g. information extraction systems often make use of it, it has some flaws: It fails in particular to reflect the core property of language being inherently sequential — the BOW encodings of "Alice hit Bob" and "Bob hit Alice" are indistinguishable.

Therefore, other methods of representing word sequences numerically have been devised, which assign to each word in a sequence a numeral representation, thus preserving the sequential information. Examples of such a representation would be Latent Semantic Analysis [Furnas et al., 1988], or word2vec [Mikolov et al., 2013] which showed the possibility of letting a neural model compute those embeddings in an unsupervised manner.

Contextualized Word Embeddings

First implementationBengio et al. [2003]

The architecture computing word embeddings that has caused the most uproar recently was probably BERT Devlin et al. [2018], an architecture that led to so many variants of it, that it created a whole new field inside the NLP community — the BERTology Rogers et al. [2020]. BERT is a good example for a typical neural age NLP model: It's architecture is completely agnostic of knowledge about language whatsoever, it "only" operates on sequental concatenations of symbols; there is also no preprocessing of the data — no POS-tagging, no dependency parsing, no NER.[5] Albeit this complete lack of any sort of explicit linguistic knowledge, by only extracting statistical patterns it learns from processing huge amounts of text, BERT achieved several SOTAs on well-established NLU data sets, such as GLUE Wang et al. [2018].

However, as I will describe in more detail in the next chapter, BERT exposes also some weaknesses and undesirable flaws: While being able to perform surprisingly

[5]Because of this non linguistic specific architecture, BERT can easily be adapted to operate on other sequential data. This has actually been done: Ji et al. [2020] for example trained a DNABERT model for successfully deciphering non-coding DNA.

well on some tasks, there are situations where BERT fails in rather trivial situations.

Semantic Structures

While the motivation for representing a sentence numerically is to let algorithms operate then "autonomously" on these representations, for example by computing some distance measures between a query and a set of texts for information retrieval, one can also try to encode some structural information of the sentence at hand. To support a model with information about what grammatical relations between parts of a sentence are present, one could provide the algorithm with the syntax parse tree of the sentence. This way, the difference between the afore mentioned two "hitting"-sentences would be clearly distinguishable.

Semantic Role Labels

Semantic Roles are a linguistic tool developed for analyzing a sentence regarding the semantic relations that hold between different entitles involved in the action described by it. The core idea is to identify generalizable semantic functions, or semantic roles, that participants in an event can engage in. With such an instrument, it is possible to model the semantic equivalence of grammatical and syntactical quite different sentences (after Palmer et al. [2010]:

(1.1) John broke the window.

(1.2) The window broke.

Although "break" is a transitive verb in the first sentence while in the second an intransitive; although "window" is the grammatical object in the prior whil the grammatical object in the latter — the expressed action is both times that an object, the window, was shattered. Semantic Roles are an attempt to formalize this by attributing each noun phrase a generalizable label. In this case, "the window" would be labelled as *patient* , i.e. the particiapnt undergoing a change of state, in both sentences. In thge first, where there is also a clear initiator of this change, "John" would be labelled as the *agent* of the event.

Thus, any sentence can be represented by the words in it replaced by their Semnantic Role Label (SRL): <John> <broke> <the> <window> → <AGENT> <PREDICATE> <PATIENT> <PATIENT>. (Or, as mentioned before, a combination of the "normal" text represenatation combined with SRLs)

1.2 Contributions/Goals

"Recently at ACL conferences, there has been an over-focus on numbers, on beating the state of the art. Call it playing the Kaggle game." [Manning, 2015, p. 702]

As I laid out before, in the past decades computational linguistics has undergone several "revolutions" which, although some people might see this differently, can be described as moving from a strong emphasis on linguistics to a more data-driven computational discipline.

Furthermore, the introduction of deep learning into computational linguistics has introduced a so called *black box*; which means essentially that although the underlying formulas and the architecture of neural nets are well-known — the mathematics behind them is rather simple —, it is nevertheless impossible to determine *what exactly* those models learn from the data.

"It would be good to return some emphasis within NLP to cognitive and scientific investigation of language rather than almost exclusively using an engineering model of research." [Manning, 2015, p. 706]

I see one of the contributions of my model also in re-introducing some sort of linguistic considerations in the current NLP efforts and also to

1.3 Research Questions

The research questions that shall be answered in this thesis, are:

1. Does enriching BERT embeddings with SRL information have a positive, measurable effect on NLU tasks?
2. What role play different implementations of SRL-enriching?

Question 1 has the following subquestions: **(a)** Since I replicate in some sense the work of Zhang et al. [2019b], are my findings similar to what they reported? **(b)** Are there differences between datsets, registers, tasks, etc.? **(c)** Are the datasets well-suited for making sound statements about the effect of enriching BERT embeddings with SRLs? **(d)** In case (c) does not hold, is there a way to determine nevertheless if SRLs might have a positive effect in more appropriate settings? **(e)** Can I determine what aspect(s) of SRLs support models in downstream tasks?

Question 2 can be broken down into: **(a)** The number of predicate-argument struc-

tures is defined to be three per sentence; is it more effective to fill empty slots with 0-SRLs, or duplicating existing SRLs? **(b)** To merge BERT embeddings with SRLs, either the splitted BERT subtokens need to be merged back or the SRLs have to splitted up — does this lead to different results?

1.4 Thesis Structure

In this first chapter I gave a very brief overview of general trends in NLP over the past decades and highlighted some

Chapter 2 introduces the basic concept of BERT, its shortcomings and presented solution or improvements. Further, I explain my approach and the relating linguistic concepts.

The datasets I compiled to test my models on are described in detail in chapter 3.

In chapter 4, I describe the details of the architecture of my BERT-variant in all detail: The identification and encoding of the semantic role labels, the different combination procedures of the BERT embeddings with these, as well as the different head architectures.

The overview and discussion of the performance of the various model architectures on the GerGLUE data set is made in chapter 5.

Finally, I draw some insights and conclusions in the last chapter 6.

2 Approach

In this chapter, I will give a brief overview of several things: In a first step, I present the BERT architecture and its impact on NLP in recent years. Secondly, I will shortly demonstrate Problems that have been identified relating to the performance of BERT. Then, I will point out some submitted solutions countering those problems. And lastly, I will describe my approach and elucidate the topic of semantic roles.

2.1 BERT

Since the publication of the seminal paper "BERT: Pre-training of deep bidirectional transformers for language understanding" Devlin et al. [2018] and the accompanying open-sourcing of its architecture[1], BERT has probably been the most studied and cited NLP model since word2vec Mikolov et al. [2013] — amassing over 17,000 citations on Google Scholar as of April 2021. This massive interest from the NLP community in BERT suggests that it somehow must be accomplishing something which is of greater significance to the field than regular benchmark SOTA cracking by "normal" new or improved architectures.

The basic concept of BERT is straight forward: (1) Let a big, sophisticated neural network learn contextualized embeddings for words by training it unsupervised on huge amounts of data. (2) Use these embeddings as representations for words in downstream tasks, put a very simple neural network on top (mostly a simple FFNN) and fine-tune them during training on the downstream task.

One of the surprising findings of Devlin et al. was the transferability of these word embeddings: Although the task of the pretraining learning has nothing to do with the downstream task, where the embeddings are used in, the BERT embeddings can be fine-tuned in a lean manner to achieve SOTA results on established NLU datasets. Another advantage of this approach is that the cost, hardware, and data intensive pre-training of the embeddings (step one) must only be computed once;

[1] `https://github.com/google-research/bert`

the downstream task dependent fine-tunig can then be carried out in a lean set up.[2]

From a more technical point of view is BERT first and foremost a neural network architecture. More precise, it is an implementation of the self-attention mechanism introduced by Vaswani et al. [2017]; the main difference, and apparently advantage, to other architectures that implement the transformer architecture is that BERT is a bi-directional language model.I will not go into too much details here since BERT is now a very well-known structure and has been described in a plethora of papers, blogs, and videos.In simple terms, BERT takes an input sequence, tokenizes it and computes contextualized vector representations for each token via stacked blocks employing self-attention and linear combination. Figure 1 shows one of these blocks.

The computation of the weight matrices and initial vector representations for the tokens is done via an unsupervised learning phase, often referred to as pre-training. The basic concept is that BERT is given sentence-chunks of large amounts of text (Devlin et al. use the BooksCorpus, consisting of 800 million words, plus the English Wikipedia, consisting of 2.5 billion words) and BERT needs to optimize on two training objectives: (1) One word is randomly masked and BERT has to predict it, and (2) BERT has to decide if, given two randomly sampled sentences, the second is a valid continuation of the first. Crucially, both tasks can be generated automatically, no tedious human annotation of data is needed.

With this "simple" approach — i.e. unsupervised pretraining of contextualized embeddings and fin-tuning on target tasks with very simple head on top — Devlin et al. beat the hitherto leading architecture on the GLUE benchmark by an outstanding average of 7,0%. This is especially remaking, since BERT is not a highly specialized model[3], but apparently still more effective on most tasks than highly task-specific

[2]To give an impression on the expenses of pre-training the BERT architecture: Schwartz et al. [2019] estimate the pre-training for BERT-large to have lasted four days on 64 TPU chips, resulting in power expenditures of about $7,000. However, this has to be considered rather cheap compared to recent architectures' sizes: The largest architecture to this date is the T-NLG (Turing Natural Language Generation) built by Microsoft, possessing a staggering 17 billion parameters — that is approximately 48 times the size of BERT-large (350 million parameters), cf. Sharir et al. [2020]. Open AI's GPT-3's [Brown et al., 2020] pretraining is estimated to have costed $12 million [Floridi and Chiriatti, 2020]. This trend of increasingly bigger language models has earned severe critique from several sides, ranging from ecological and social to linguistic concerns over such models; for a good overview of these points see Bender et al. [2021].

[3]Until then, SOTAs were achieved by complex interwiring of some embeddings with a specialized architecture.

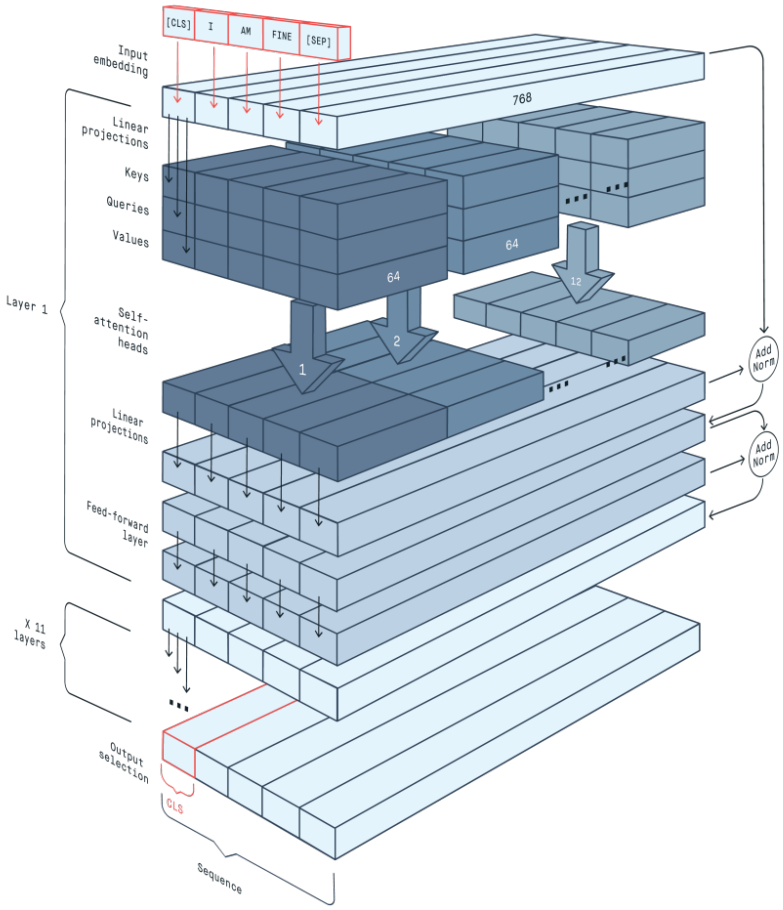


Figure 1: 3D-visualization of the BERT architecture. Nicely illustrated are the 12 attention heads and following linear projections in one block (here called "Layer"). Credit for the image goes to Peltarion

optimized models. Thus, the NLP community was awestruck.

2.1.1 Problems

In recent years, a lot of research went into analyzing, improving, and deluding the BERT architecture; in the NLP field, those efforts are often referred to by the notion of "BERTology" (cf. Rogers et al. [2020]). While BERT showed to be astonishingly effective on several established data sets and benchmarks such as GLUE [Wang et al., 2018], it soon became obvious that it also had its weak-spots: Ettinger confronted BERT with three language tasks originally coming from psycholinguistics which are well-known to be difficult to tackle, even for humans. They find

> that [BERT] shows sensitivity to role reversal and same-category distinctions, albeit less than humans, and it succeeds with noun hypernyms, but it struggles with challenging inferences and role-based event prediction—and it shows clear failures with the meaning of negation." [Ettinger, 2020, p. 46]

Apparently, while being able to solve "regular" tasks, BERT seems to be prone to fail in situations where proper semantic understanding of text sequences is crucial, such as detecting role reversal or sentence completion tasks.

Jiang and de Marneffe [2019] also state "that despite the high F1 scores, BERT models have systematic error patterns", which for them suggests "that they still do not capture the full complexity of human pragmatic reasoning."

Jin et al. [2020]even go further and created so-called adversarial attacks on BERT: After observing that BERT seems to rely only on the statistical cues of a small number of the input tokens to form its predictions, Jin et al. create a sophisticated algorithm to permute the input sentences wihtout actually changing its meaning. Following is an example from the SNLI [Bowman et al., 2015] dataset, exemplifying one such attack:

(2.1) **Premise**: A child with wet hair is holding a butterfly decorated beach ball.

Original Hypothesis: The *child* is at the *beach*.
Adversarial Hypothesis: The *youngster* is at the *shore*.

The italicized words are the ones that were affected by the adversarial algorithm. Obviously — for a human — the meaning of the adversarial hypothesis has not changed from the original one. One could argue that there is a slight difference in style (the adversarial sounds somewhat overblown to me) but it would still count as an entailment of the premise. However, as the authors report, BERT is affected by such attacks and changes its predictions.

For SNLI, Jin et al. report to bring down BERT's original accuracy of 89,4% to an astonishing 4,0% by permuting 18,5% of the input tokens. Recall that the permutations are, simply put, nothing else than exchanging words with identical meanings.

2.1.2 Solutions / Related Work

But the NLP community was not only passive and/or destructive concerning BERT, it also produced a vast number of adaptions, variations, and improvements to the "vanilla"-BERT. The motivations behind all those BERTlings are as manifold as one

can think: some are simply adaptions to other languages than English, some are general variations (in hope of improvement) of the BERT architecture, other are explicitly addressing above outlined problems.

The following overview sheds some light on the iridescent potpourri of the BERT family.[4] Note however that this is by no means an exhaustive presentation of all BERT-variants produced so far.

Adapting to other languages One of the most straightforward modifications to the BERT model is to pre-train it on different langauges. Examples for this are the French CamemBERT Martin et al. [2019], the Italian AlBERTo Polignano et al. [2019], or the Dutch BERTje de Vries et al. [2019].[5]

Adapting to specialized domains BERT was pre-trained on two corpora: (1) The BooksCorpus [Zhu et al., 2015], consisting of 800 million words and comprising 16 different genres. (2) The English Wikipedia, lists and tables excluded (2.5 billion words). Examples of BERTs pre-trained on specialized domains are for example BioBERT Lee et al. [2020], which is an adaption to biomedical language, and LEGAL-BERT Chalkidis et al. [2020] which is a whole family of BERT models pre-trained on legal texts.

Including different modalities Another, highly interesting amplification of the BERT architecture is the inclusion of additional modalities, for exmple (moving) images: VideoBERT Sun et al. [2019a] learns embeddings for image-enriched texts and can be used for example for image captioning or image classification tasks. Several researchers claim that the future of NLP relies on combining text with sensory, e.g. visual, data for creating more stable and reliable models; cf. [Bisk et al., 2020; Bender et al., 2021].

Optimizing architecture/training objective(s) Several BERT-variations modify the actual architecture of BERT: DistilBERT Sanh et al. [2019] is a variant 60% of the size of the original BERT while retaining 97% of its original performance. RoBERTa Liu et al. [2019] essentially modifies core hyper-parameteres such as batch-size, byte-level BPE, and the like, creating a more stable BERT. DeBERTa He et al. [2020] modifies the attention mechanism and the position encoding, while TransBERT Li et al. [2021] introduces a new pre-training

[4] This compilation is partly drawn from the towards-data-science article "A review of BERT based models" by Ajit Rajasekharan.

[5] Devlin et al. [2018] also trained a multi-lingual BERT (mBERT), which was trained on 104 languages. However, language-specific BERTs have been shown to be more performant than employing the mBERT.

framework.

Incorporating structured information One of the strengths of BERT is the unsupervised pre-training on unstructered, raw text. However, research has shown that including structured linguistic information can stabilize BERT and even counterbalance some of the known weaknesses (see above) to some extent: ERNIE Sun et al. [2019b] includes a knowledge graph into BERT, making structural fact representations available to BERT. VGCN-BERT Lu et al. [2020] combines a Vocabulary Graph Convolutional Network with the standard BERT and Zhang et al. [2019b] include semantic role labels in their SemBERT during fine-tuning.

2.2 GliBERT

Infected by the pandemic BERT-fever and slightly annoyed by the hegemony of English, I decided to enqueue in this illustrious list by combining two of the advancement startegies: I try to improve the German BERT by enhancing it with structured, linguistic information during fine-tuning.

Since it is common practice to give your enhanced/variegated BERT architecture an appropriate name — you may have spotted a pattern in the sections before — I decided to not deviate from this tradition, and decided to call my breed **G**erman **l**inguistic **i**nformed BERT, or short: **GliBERT**. And thus a new

There exist several linguistic structures which could be hypothetically included into BERT: GermaNet [Hamp and Feldweg, 1997] is a large lexical-semantic net that relates noun, verbs, and adjectives semantically by grouping lexical units that express the same concept into synsets and by defining semantic relations between these synsets. It can also be characterized as a thesaurus or a light-weight ontology.

Another, much simpler, possibility would be to identify named entities and include the encoded structured information related to them, e.g. using the DBpedia [Auer et al., 2007], and enrich the BERT embeddings with them.

However, I decided to concentrate on semantic roles for several reasons: Zhang et al. demonstrated the feasibility of this undertaking for English. Further, it occurred to me to be a good balance between two extremes: (1) Including sophisticated knowledge structures, which would require extensive preprocessing (stemming, identifying content words, potential word sense disambiguation, look-up in the knowledge base) and rather cumbersome encoding; and (2) stright-forward on the fly “mark-up” of

input text, with low information substance in the case of named entities. with semantic roles, I get the best from both worlds: Easy to implement structured information, while — hopefully — truly adding semantic substance to the vanilla BERT embeddings.

All code relating to the following dataset set ups, GliBERT architecture, and training can be found in my GitHub repository.

2.3 Semantic Roles

One difficulty a system targeted at NLU must tackle is the ability to cope with the vast amount of flexibility and freedom in natural language to express things. In NLU, often one would probably talk about propositions: Facts about the world are being stated and properties of this propositions need to be understood or processed in some way or the other. Problems now arise due to t

(2.2) Due to the big waves, the ship was severely damaged and went down.

(2.3) The stormy ocean caused the vessel to sink.

(2.4) Unable to save the stricken freighter, the crew had to be evacuated.

Although these three sentences make use of very different vocabulary — an unweighted BLEU score is virtually zero between them — it is obvious to a speaker of English that they all convey more or less the same meaning, that all of them tell the same state of affairs: A ship sank because of the forces of nature.

Maybe the most obvious way of saying the same thing with different words is synonymy: “Ship”, “vessel”, and “freighter” all refer to the same object in the examples above; having several options when choosing a word to denote something is a paramount feature of human language.

Further, the communication of one and the same event, e.g. the sinking of a ship, can be transmitted in various ways: In the second sentence, the process is denominated explicitly using the verb “to sink”; in the first, the semantically more obscure semi-fixed expression “to go down” is used to inform about that very situation; while in the third the sinking of the ship is not mentioned explicitly but inferable from the circumstance of “not being able to save” it.

For a human speaker, all this disentangling, recognizing coreference, reconstructing not explicitly mentioned context, etc. happens effortless and automatic — for an algorithm, however, phenomena like the ones mentioned before pose serious chal-

lenges.

"For computers to make effective use of information encoded in text, it is essential that they be able to detect the events that are being described and the event participants." [Palmer et al., 2010]

As I laid out in section 2.1.1, a modern, unsupervised model like BERT seems to perform surprisingly good in tasks where such "understanding" of events are being tested.[6] Simultaneously, some investigated failures of BERT seem to indicate that this "understanding" goes not too deep.

Semantic Roles are an attempt at creating an instrument with which it is possible to analyze sentences as 2.3 and capture the semantic similarity between them. The central idea hereby is that every utterance has an underlying semantic structure[7] (sloppily phrased: "Who did What to Whom, and How, When and Where?") which can be realized in different surface structures. There have been various undertakings in creating a vocabulary for describing such structures, putting the focus on different aspects and showing varying degrees of analytic detail.

The paper "The Case for Case" [Fillmore, 1967] is often seen as the starting point for the theory of semantic roles in modern linguistics. Fillmore argued in it that what he called "Deep-Cases" play a crucial role in the Deep-Structure of sentences — the hitherto prevalent view in Generative Grammar was that case was a purely Surface-Structure related phenomenon and only one of several possibilities to realize syntactic relationships. Interestingly, these "Deep-Cases" were semantically-motivated; he proposed seven Deep-Cases, e.g. the "Agentive": " [T]he case of the typically animate perceived instigator of the action identified by the verb", or the so-called "Factitive": " [T]he case of the object or being resulting from the action or state identified by the verb, or understood as a part of the meaning of the verb" [Fillmore, 1967, p. 46]. The observable Surface-From cases could than through elaborate tests give insights as to what the underlying Deep-Cases were.

Building upon those core concepts introduced by Fillmore, other linguists added features to the project of formalizing the core semantic structures found in language, as summarized by Palmer et al.: In the beginnings of the 70ies, Jackendoff [1972] expanded and refined Fillmore's model by introducing the concept of primitive conceptual predicates and their property of governing arguments, which were

[6] Of course, one does not really measure epistemical understanding in such tests, but this is maybe the closest we get (cf. Sahlgren and Carlsson [2021])

[7] Often, especially in Generative Grammar traditions, this level is also known as deep structure, or D-structure.

conceptualized as bearing some proto-semantics, similar to the Fillmorian Deep-Cases. This approach, known as "Lexical Conceptual Structure" (LCS), proved to be an elegant theory and capable of generalizing well between different verbs; in the 90ies LCS was implemented as system for representing semantics in early NLU and translation models [Palmer et al., 2010]. But, due to its detailed analysis of verbs into (several) primitive predicates and the highly verb specific conceptualized semantic roles of them, LCS turned out to be cumbersome to extend to the whole range of a vocabulary of a language.

Dowty [1991] in contrast, approached the problem of constructing a framework for analyzing core conceptual semantic structures from a different angle: Instead of providing a detailed description of the primitive predicate and idiomatic argument structure for each individual verb, he attempted to identify general functions of noun phrases, what he called "thematic proto-roles", present in sentences. To accomplish this, Dowty drew from the theory of "family resemblance" and defined a set of attributes which would indicate such a thematic role. "The hypothesis put forth here about thematic roles is suggested by the reflection that we may have had a hard time pinning down the traditional role types because role types are simply not discrete categories at all, but rather are cluster concepts [...]" [Dowty, 1991, p. 571]

Proto-Agent properties (after [Dowty, 1991, p. 572]):

a volitional involvement in the event or state

b sentence (and/or perception)

c causing an event or change of state in another participant

d movement (relative to the position of another participant)

e (exists independently of the event named by the verb)

I will not elaborate this further, but other theories have been put forward.

Like many theories in linguistics, Semantic Roles remain a disputed topic in the field until today: "There may be general agreement on the cases (or Thematic Roles or Semantic Roles) [...], but there is substantial disagreement on exactly when and where they can be assigned and which additional cases should be added, if any." [Palmer et al., 2010]

However, different resources have been created, implementing some variety of the various Semantic Roles frameworks. One of these is the PropBank [Kingsbury and Palmer, 2002].

Semantic Roles are systematic abstractions of semantic functions that are attributed to the participants in a factual situation: The volitional acting entity in a situation is abstracted as "(Proto-)Agent"; regardless of the actual, concrete act. Similarly, noun phrases which denote participants that

"Because verbs generally provide the bulk of the event semantics of any given sentence, verbs have been the target of most of the existing two million words of PropBank annotation. Nonetheless, to fully capture event relations, annotations must recognize the potential for their expression in the form of nouns, adjectives and multi-word expressions, such as Light Verb Constructions (LVCs)." [Bonial et al., 2014, p. 3014]

(2.5) He fears bears.

(2.6) His fear of bears.

(2.7) He is afraid of bears.

Bonial et al. [2012] define the following proto-roles for the numbered arguments, modifiers, and relations:

Arg0 agent

Arg1 patient

Arg2 instrument, benefactive, attribute

Arg3 starting point, benefactive, attribute

Arg4 ending point

ArgM modifier

Rel Relation (can be a verb, noun, or adjective)

Following are some exmample sentences from the PropBank frames[8]. Semantic roles are highlighted using the colors from the previous list. Note that only one relation is marked in the sentences, even if there are multiple. Since DAMESRL only treats verbs as semantic roles distributing relations, I include only verbal "Rel"s in the examples:

(2.8) [$_{\text{Arg0}}$ Yasser Arafat] has [$_{\text{Rel}}$ written]
[$_{\text{Arg2}}$ to the chairman of the International Olympic Committee], asking him

[8] accessible through this GitHub repository

to back a Palestinian bid to join the committee.

(2.9) Once [Arg0 he] [Rel realized] [Arg1 that Paribas's intentions weren't friendly], he said, but before the bid was launched, he sought approval to boost his Paribas stake above 10%.

(2.10) [Arg1 National Market System volume] [Rel improved] [Arg4 to 94,425,00 shares] [Arg3 from 71.7 million Monday].

(2.11) [Arg0 The new round of bidding] would seem to [Rel complicate] [Arg1 the decision making] [Arg2 for Judge James Yacos].

(2.12) The action followed by one day an Intelogic announcement that it will retain [Arg0 an investment banker] to explore alternatives "to [Rel maximize] [Arg1 shareholder value]," including the possible sale of the company.

(2.13) [Arg0 He] [ArgM-mod would] scream and [Rel cut] [Arg1 himself] [Arg3 with rocks].

NOTE: [9]

"The main reason computational systems use semantic roles is to act as a shallow meaning representation that can let us make simple inferences that aren't possible from the pure surface string of words, or even from the parse tree." [Jurafsky and Martin, 2019, p. 375]

In the literature, often Gildea and Jurafsky [2002] is considered to have formally defined the task of automatic SRL.

"Analysis of semantic relations and predicate-argument structure is one of the core pieces of any system for natural language understanding." [Palmer et al., 2010]

PropBank Roles, according to Bonial et al. [2012]:

Thanks to lexical resources such as the PropBank, a multitude of models aiming at labeling sentences with semantic roles are now available. DAMESRL [Do et al., 2018], trained on the CoNLL '09 [Hajič et al., 2009] data (which implements PropBank style SRLs).

Following some examples of DAMESRL-labelled sentences stemming from the GliB-

[9] To me, not all annotations in PropBank are beyond all doubt; for example, in sentence 2.12 "an investment banker" is labelled as agent "maximzing" the the patient "shareholder value" — however, I would argue that it's rather the "alternative" that take proto-agentive role in maximizing the shareholder values.

ERT corpus (see chapter 3). The sentences are represented vertically with the left-most column being the actual sentence; each column represents one identified verb (B-V) and its predicted semantic roles, labelled using the BIO-schema[10]

deISEAR

Ich	B-A0	O
fühlte	B-V	O
[MASK]	B-A1	O
,	I-A1	O
als	I-A1	O
ich	I-A1	B-A0
aus	I-A1	O
Versehen	I-A1	O
schlechte	I-A1	B-A1
Milch	I-A1	I-A1
getrunken	I-A1	B-V
habe	I-A1	O

MLQA

Welche	B-A1	B-A2
Positionen	I-A1	I-A2
muss	O	O
man	B-A0	B-A0
erreichen	B-V	O
,	O	O
um	O	O
die	O	B-A1
von	O	I-A1
Kaius	O	I-A1
angeordnete	O	I-A1
Position	O	I-A1
eines	O	I-A1

[10]Introduced by [Ramshaw and Marcus, 1999], the BIO-schema is a convenient way of adding a label to each token in a sequence, indicating if it belongs to a certain subgroup, or chunk, of the sequence. For example, to mark the prepositional phrase in a syntagma like "He is running from the bear", one would mark the word beginning the PP with **B**, any other words inside the PP with **I**, and all other words outside of it, using **O**: "He[O] is[O] running[O] from[B-PP] the[I-PP] bear[I-PP]".

Läufers	O	O
einzunehmen	O	B-V
?	O	O

XNLI

Es	O	O	O
war	O	O	O
das	O	O	O
Wichtigste	O	O	O
was	B-A1	O	O
wir	B-A0	O	O
sichern	B-V	O	O
wollten	O	O	O
da	O	O	O
es	O	O	O
keine	O	B-A1	O
Möglichkeit	O	I-A1	O
gab	O	B-V	O
eine	O	B-A1	B-A3
20	O	I-A1	I-A3
Megatonnen	O	I-A1	I-A3
-	O	I-A1	I-A3
H	O	I-A1	I-A3
-	O	I-A1	I-A3
Bombe	O	I-A1	I-A3
ab	O	I-A1	O
zu	O	I-A1	B-A5
werfen	O	I-A1	B-V
von	O	I-A1	I-A5
einem	O	I-A1	I-A5
30	O	I-A3	I-A5
,	O	O	O
C124	O	O	O
.	O	O	O

3 Data Sets

3.1 GerGLUE

Because semantics is such a fuzzy, hard to formalize property of language, it is not easy to assess the capabilities of an architecture designed at solving problems related to meaning. In the data-driven NLP community today, it is common practise to measure the power of a model by measuring it's performance on some standardized data set. However, a model aiming at capturing semantics of human langauge "must be able to process language in a way that is not exclusive to a single task, genre, or dataset", as Wang et al. [2018] correctly point out.

To provide a standardized set of data sets for the NLP community to compare different NLU-committed models, Wang et al. compiled the General Lanuage Understanding Evaluation, short GLUE, benchmark. It consists of nine data sets addressing different NLU problems; from acceptability tasks (is the phrase "Saw the man the dog." an acceptable English sentence?) to detecting textual entailment (is the meaning of "A boy is at the beach" enatiled by the sentence "Two kids are building a sandcastle at the beach"?). See table 1 for a list of all GLUE data sets, their tasks and

Following Wang et al. [2018],

[1]Wang et al. [2018] reformulate the original SQuAD task CITE of predicting an answer span in the context into a sentence pair binary classification task: They pair each sentence in the context with the question and predict whether or not the context sentence includes the answer span.

[2]Wang et al. [2018] combine several data sets into RTE; for data sets that have three labels — *entailment*, *neutral*, and *contradiction* — they collapse the latter two into one label *not_entailment*.

[3]In the original Winograd Schema Challenge CITE, the task is to choose the correct referent of a pronoun from a list. Wang et al. [2018] reformulate this to a sentence pair classification task, where the original sentence is paired with the original sentence with each pronoun substituted from the list and then predicting whether the substituted sentence is entailed by the original one.

Data Set	NLP Task	ML Task	# Examples	Splits
	Single-Sentence Tasks			
CoLA	Acceptability	Binary Classification	8.5k/1k	train/test
SST-2	Sentiment Analysis	Binary Classification	67k/1.8k	train/test
	Sentence Pair Tasks			
MNLI	Natural Language Inference	Multi-Class Classification	393k/20k	train/test
MRPC	Paraphrase Identification	Binary Classification	3.7k/1.7k	train/test
QNLI	Question Answering	Binary Classification[1]	105k/5.4k	train/test
QQP	Paraphrase Identification	Binary Classification	364k/391k	train/test
RTE	Natural Language Inference	Binary Classification[2]	2.5k/3k	train/test
STS-B	Sentence Similarity	Regression (1 - 5)	7k/1.4k	train/test
WNLI	Coreference Resolution	Binary Classification[3]	634/146	train/test

Table 1: Original GLUE data sets and tasks (following the table from Wang et al. [2018]).

Data Set	NLP Task	ML Task	# Examples	Splits
	Single-Sentence Tasks			
deISEAR	Emotion Detection	Multi-Class Classification	1,001	-
SCARE	Sentiment Analysis	Multi-Class Classification	1,760	-
	Sentence Pair Tasks			
MLQA	Question Answering	Span Prediction	509/4,499	dev/test
PAWS-X	Paraphrase Identification	Binary Classification	49,402/2,001/2,001	train/dev/test
XNLI	Natural Language Inference	Multi-Class Classification	2,489/5,009	dev/test
XQuAD	Question Answering	Span Prediction	1,179	-

Table 2: GerGLUE data sets and tasks.

3.1.1 General Issues

There are a few remarks and strategies that apply to all collected corpora:

(1) All of the data sets except deISEAR are not monolingual, i.e. German, sources, but bi- or multilingual corpora. To compile a German GLUE corpus I only use the German subset of those corpora. For example, the MLQA data set provides all 49 combinations of the languages it contains: Context in Arabic, question in Hindi; context in English, question in Spanish, etc. Also in this case, I choose only the German-German part of the data set for my corpus.

(2) The data sets I chose for my little GLUE corpus are being provided in different modes. While three of the corpora, namely MLQA, PAWS-X, and XNLI, come with

a predefined split, the others are made available without splits. In the latter case, I split the data sets into train, development, and test splits using a 0.7, 0.15, and 0.15 portion, respectively. Interestingly, the data sets that come with splits, only provide a development and test portion. To ensure that my results are comparable with those that the authors of the different data sets report, I leave the test split as it is, and split the development set into a train and development set, implementing a 85:15 ratio.

(3) Most of the data sets were constructud by translating existing monolingual English data sets (semi-)automatically into the different target languages. As I show in section REFXXXX, this does not come without introducing noise into the data.

The following differences to the original GLUE corpus must be noted:

(1) While Wang et al. [2018] reformulate a multitude of tasks into inference tasks, I follow in my implementation Zhang et al. [2019b] and approach the question answering tasks as Devlin et al. [2018] in the original BERT implementation; i.e. as span prediction task.

(2) I tried to combine a multitude of different tasks into my GLUE dataset (single sentence tasks vs bi- or multiple sentence tasks, classification vs. span detection, different semantic problems such as emotion detection, question answering etc.), I could not compile all tasks that appear in GLUE into my semantic dataset compilation. For example, there are data sets that concern linguistic acceptability in the original GLUE corpus, such as e.g. CoLA Warstadt et al. [2019], or XXX . To disregard this task was not an intentional decision, but due to fact that there are simply not as many datasets available for German and apparently there are no datasets addressing linguistic acceptability in German.

3.2 Corpora

In this section, I give a detailed description of the selected data sets in alphabetical order: What kind of task is addressed, what is the text variety, and report some statistical measures, e.g. the average length of examples in the different sub-sets (Training, Development, Test).

3.2.1 deISEAR

This data set addresses the task of Emotion recognition, a sub-task of Sentiment Analysis. Technically, it is a sequence classification problem: Given a sequence of tokens $x_1 \dots x_n$, predict the correct label y from a fixed set of emotions Y. Or. in a more natural way of speaking, what emotion expresses a ceratin statement? Following the original study "International Survey on Emotion Antecedents and Reactions" [Scherer and Wallbott, 1994], Troiano et al. [2019] constructed their data set for German:

In a first step, the authors presented annotators with one of seven emotions, and asked them to come up with a textual description of an event in which they felt that emotion. The task was formulated as a sentence completion problem, so the annotators, which were recruited via a crowdsourcing platform, had to complete sentences having the following structure: "Ich fühlte *emotion*, als/weil/dass ...". Seven emotions were given for which the descriptions had to be constructed: Traurigkeit, Ekel, Schuld, Wut, Angst, Scham, Freude.[4]

The second phase of the data generation process comprised of re-labeling the generates sentences such that five annotators annotate each sentence. The emotion word was omitted, and the annotators had the list of seven emotions at hand. The authors report that for approximately half of all sentences, the inter-annotator agreement was perfect; i.e. each of the five annotators attributed the same emotion. However, some emotions seem to be prone for not being clearly separable: Shame, for example, gets confused with guilt in 35% of the cases. The authors do not report any coefficient, but since they provided the complete results of phase 2, I was abe to compute the Fleiss' κ, a standard metric for estimating annotator agreement, and thus, reliabilty of these labels; the computed value equals to 0.66, which corresponds to "substantial" agreement in the interpretation scale for th variable, proposed by Landis and Koch [1977].

Following are seven example sentence randomly picked out of the deISEAR corpus, one for each emotion.

(3.1) Ich fühlte **[Traurigkeit]**, als mein Laptop kaputt ging und die Garantie schon abgelaufen war.

[4]Interestingly, out of these seven emotions, six represent rather negative emotions — only *Freude* is a clearly positive sensation. Maybe negative emotions are more lucidly detactable (by humans and/or machines) than positive ones, I leave this question for the reader to further explore.

(3.2) Ich fühlte **[Scham]**, weil mir mal beim Urlaub das Geld ausging.

(3.3) Ich fühlte **[Freude]**, als ich mit meinen Arbeitskollegen ohne Ende Witze gerissen habe.

(3.4) Ich fühlte **[Angst]**, als der Chef sagte dass Mitarbeiter gekündigt werden müssen.

(3.5) Ich fühlte **[Wut]**, als ich die Nachricht gelesen habe, dass der VfB Stuttgart nicht in neue Spieler investieren wird.

(3.6) Ich fühlte **[Ekel]**, als ich verschimmeltes Essen im Kühlschrank gefunden habe.

(3.7) Ich fühlte **[Schuld]**, dass ich meinen besten Kumpel versetzt habe.

Now it is up to you: Here are four sentences with masked emotions — try to assign what you thin is the correct one. The possible emotions are **Angst, Ekel, Freude, Scham, Schuld, Traurigkeit, Wut**.[5]

(3.8) Ich fühlte **[?]**, als ich meine kleine Tochter zum Schwimmen abgeholt habe.

(3.9) Ich fühlte **[?]**, als meine Mutter mich zur Schule begleiten musste als ich die schule geschwänzt hatte

(3.10) Ich fühlte **[?]**, als die Ärzte im KH bei meiner im sterben liegenden Großmutter einen künstlichen Zugang legen wollten um die Schilddrüsenmedikamente zu verabreichen.

(3.11) Ich fühlte **[?]**, als eine Feuerwerksrakete in Richtung meiner Kinder abgefeuert wurde und mein kleiner weinend davon lief.

3.2.1.1 Statistics

deISEAR is one of the data sets that are made available without any pre-defined training/development/test splits. Therefore I shuffle all 1,001 sentences with a 30:15:15 ratio, resulting in a training set of 700, a development set of 150 and a test set of 151 sentences.

In figure **??**, the length of the sentences and the label distributions in the three data sets are plotted. While the data sets were created randomly, there are some peculiarites observable: The lengths of the sentences in the trainig set show a greater

[5] 3.8: Freude, 3.9: Scham, 3.10: Wut, 3.11: Angst

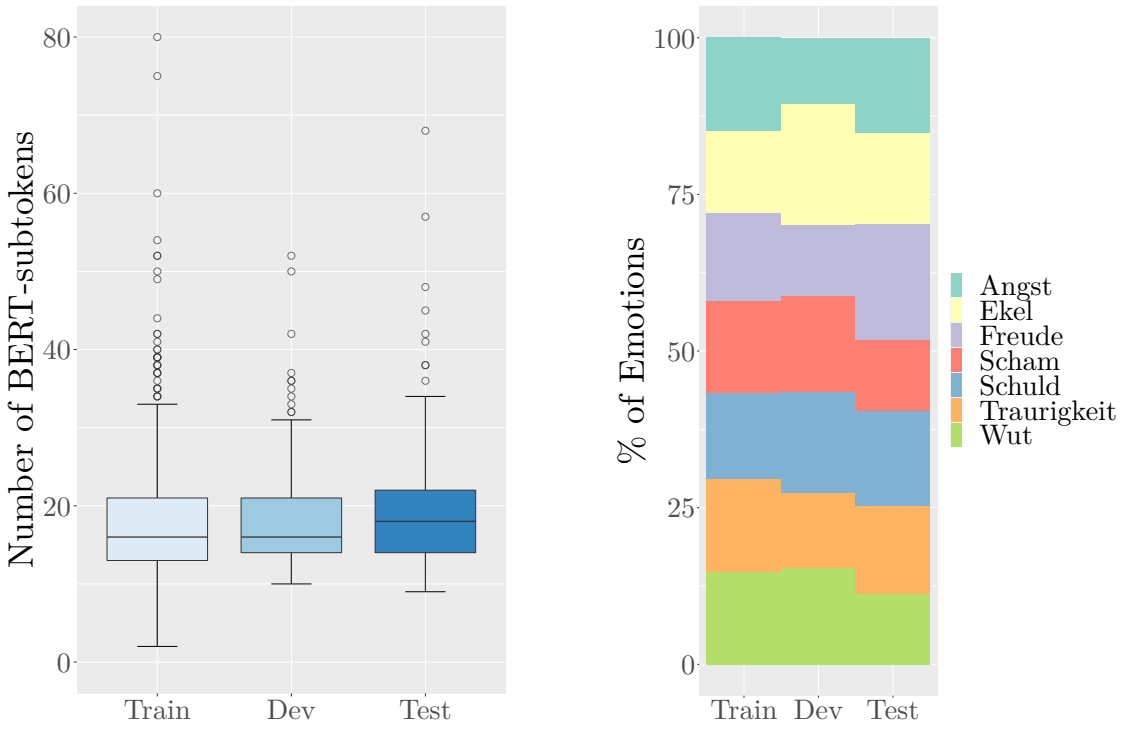


Figure 2: **Left**: Length of subtokenized deISEAR sentences. Note that one extreme outlier in the development set comprising 300 BERT-subtokens is not included in the plot. **Right**: Label distributions of deISEAR datasets.

variation compared to the development nd test set than one would except.

3.2.1.2 SOTA

Troiano et al. [2019] train a maximum entropy classifier with L2 regularization with boolean unigram features on the original ISEAR corpus (7665 instances). Since the original ISEAR study and data collection was carried out in English, they then machine translate the 1,001 deISEAR examples and evaluate on them. Using this strategy, the authors accomplish an average micro F_1 of 47. (Note: micro F_1 in settings where each example gets exactly one label assigned is the same as accuracy)

3.2.2 MLQA

The term question answerin comprises several related tasks or problems: In it's most general form, question answering refers to the ability to give a meaningful answer to any possible inquiry. This is normally referred to as *open-domain question answering* and models addressing this task require a whole pipeline of algorithms

in the background to produce acceptable results (compare Chen and Yih [2020]). Another form is so called *multi-choice question answering*, where the task for the model is to select the correct answer out of a list of options given the answer (compare Welbl et al. [2017]). The subsort of question answering MLQA addresses is so called *span prediction question answering*. The goal here is to extract the correct answer span out of a context text given the question.

Following are five random examples out of the MLQA data set:

(3.12) **Context**: Rita Sahatçiu Ora (* 26. November 1990 in Priština, SFR Jugoslawien) ist eine britische Sängerin und Schauspielerin kosovarischer Herkunft. Von 2010 bis 2016 stand sie bei Jay Z und Roc Nation unter Vertrag. Seit 2017 steht sie bei Atlantic Records unter Vertrag.

Question: Wann wurde Rita Sahatçiu Ora geboren?
Answer: 26. November 1990

(3.13) **Context**: Während Somalia an militärischer Stärke gewonnen hatte, wurde Äthiopien aufgrund innenpolitischer Umstände geschwächt. 1974 hatte die Derg-Militärjunta den abessinischen Kaiser Haile Selassie gestürzt, sich aber bald in interne Machtkämpfe verstrickt, woraufhin es zu Unruhen kam. In verschiedenen Landesteilen waren Derg-feindliche und separatistische Kräfte aktiv. Das regionale Machtgleichgewicht hatte sich zugunsten Somalias verschoben.

Question: Zu wessen Gunsten verlagerte sich die Balance of Power?
Answer: Somalia

(3.14) **Context**: Das Johnston-Atoll verlassend, drehte John nach Nordwesten ab und begann sich erneut zu intensivieren, als die Windscherung nachließ. Am 27. August Ortszeit erreichte John einen sekundären Höhepunkt mit Windgeschwindigkeiten von 210 km/h. Kurz darauf überquerte John die Datumsgrenze bei etwa 22° nördlicher Breite und gelangte in das Beobachtungsgebiet des Joint Typhoon Warning Center (JTWC) auf Guam. Durch seinen Aufenthalt im westlichen Pazifischen Ozean wurde Hurrikan John zum Taifun John. Kurz nach dem überschreiten der Datumsgrenze schwächte sich John wieder ab und die Vorwärtsbewegung kam fast zum Stillstand. Am 1. September hatte sich Taifun John zum tropischen Sturm abgeschwächt und veränderte seine Position knapp westlich der Datumsgrenze kaum. Dort blieb der Sturm die nächsten sechs Tage, während der John eine mehrere Tage andauernde Schleife entgegen dem Uhrzeigersinn zog, bis am 7. September ein Trog in die Gegend gelangte und John schnell

nach Nordosten abzog. Am 8. September überquerte John die Datumslinie wieder nach Osten und gelangte erneut in den Zentralpazifik.Dort angelangt erreichte John seinen tertiären Höhepunkt mit Windgeschwindigkeiten von 145 km/h als starker Kategorie-1-Hurrikan, ein gutes Stück nördlich der Midwayinseln. Der Trog nahm die Struktur von John auseinander und das kalte Wasser des nördlichen Zentralpazifiks tat sein Übriges. Am 10. September wurde die 120. Sturmwarnung zu John ausgegeben, mit der das System als außertropisch erklärt wurde, etwa 1600 km südlich von Unalaska.

Question: Wo wurde John zum Taifun?
Answer: westlichen Pazifischen Ozean

To demonstrate that this is by no means a trivial task — at least for us humans —, try to identify the correct answer[6] span for the following context-question pair (to be "fair" and have the same conditions for the model and you, you would only be allowed to read the context once...):

(3.15) **Context**: Das britische Parlament genehmigte Königin Victoria, ihrer Tochter als Mitgift 40.000 Britische Pfund (in heutiger Kaufkraft 3.662.803 Pfund) zu zahlen und legte die jährliche Apanage der Prinzessin auf 8000 Pfund fest. König Friedrich Wilhelm IV. gewährte seinem Neffen ein jährliches Einkommen von 9000 Talern. Das Einkommen des Prinzen war damit nicht ausreichend, um die Kosten eines standesgemäßen Haushaltes zu decken, und einen Teil der Haushaltskosten würde zukünftig Prinzessin Victoria aus ihrem Vermögen tragen müssen. Der zukünftige Hofstaat des jungen Paares wurde von der preußischen Königin und der zukünftigen Schwiegermutter Prinzessin Auguste ausgewählt. Die beiden Frauen entschieden sich überwiegend für Personen, die bereits länger im Hofdienst standen und damit deutlich älter als das prinzliche Paar waren. Prinz Alberts Bitte, seiner Tochter doch wenigstens zwei gleichaltrige und britische Hofdamen zu gewähren, wurde nicht entsprochen. Als Kompromiss wurden mit den Komtessen Walburga von Hohenthal und Marie zu Lynar zwei Hofdamen gewählt, die Prinzessin Victoria wenigstens altersmäßig entsprachen. Immerhin konnte Prinz Albert Ernst von Stockmar, den Sohn seines jahrelangen Beraters Christian Friedrich von Stockmar, als persönlichen Sekretär der Prinzessin durchsetzen. Prinz Albert, der überzeugt davon war, dass der preußische Hof die Einheirat einer britischen Prinzessin als Bereicherung und Ehre ansähe, bestand außerdem darauf, dass

[6] überwiegend antibritisch und prorussisch

> Prinzessin Victoria den Titel einer Princess Royal of the United Kingdom of Great Britain and Ireland beibehielt. An dem überwiegend antibritisch und prorussisch eingestellten preußischen Hof löste dieser Schritt allerdings nur Verärgerung aus.Der Hochzeitsort war Anlass für weitere Meinungsverschiedenheiten. Für das preußische Königshaus war es selbstverständlich, dass ein Prinz, der als zweiter in der Thronfolge stand, in Berlin heiratete. Letztlich konnte sich aber Königin Victoria durchsetzen, die als regierende Monarchin für sich in Anspruch nahm, ihre älteste Tochter in ihrem Land zu vermählen. Das Paar trat schließlich am 25. Januar 1858 in der Kapelle des St James's Palace in London vor den Traualtar.
>
> **Question**: Was war die Position der Berliner Gerichts gegenüber Großbritannien und Russland?

Lewis et al. [2019] compiled this data set using Wikipedia articles. First, they "automatically identify sentences from [...] articles which have the same or similar meaning in multiple languages."[7] Secondly, they crowdsourced questions for the english paragraphs, let them humanly translate into the target languages, and finally annotate the answer spans in the corresponding paragraphs.[8]

3.2.2.1 Statistics

MLQA is a set that comes with pre-defined development and test splits (the reason that there is no training set lies in the fact that MLQA, like most other data sets used in my thesis, are targeted for multi-lingual models. This means, that the training set is normally ...) The test set contains 4,499 examples, while the dev set is made up of 509 examples. As for all the other pre-splitted data sets, a training set is not part of the splits, this has to do with the approach how Lewis et al. [2019] intend to use their data set; see chapter 3.2.2.2 for the description of their training and evaluation process. In a first set up, I treat the MLQA development set as training set and split off a 15% portion of it as development set, resulting in 432 training instances and 77 development examples. In a second set up I shuffle the pre-defined splits and create a more suiting 70:15:15 set ratio; resulting in the following numbers of examples per set: Training 3,506, development 751, test 751.

[7] However, taking the sometimes huge contexts into account, I think the better formulation would have been "paragraphs" with similar meaning, instead of "sentences".

[8] If this was done manually or by implementing some other techniques is, unfortunately, not reported. The presence of sometimes strange offsets (included commas, misssing prepositions as in example 3.14 etc.) seem to indicate a not fully hand-made annotation — at least to me.

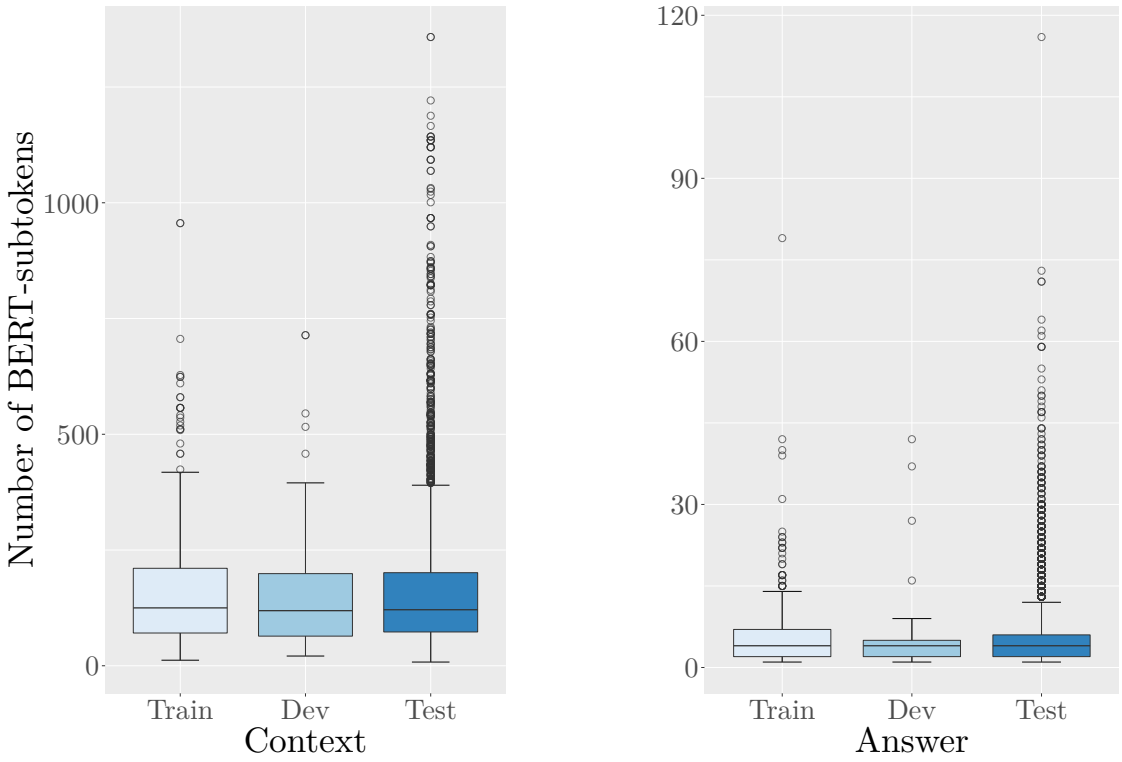


Figure 3: Left: Length of subtokenized MLQA contexts. **Right**: Length of subtokenized MLQA answers. Note the difference in y-axis scaling between the two plots: The contexts are longer by orders of magnitude to the answer spans. In fact, most answer spans consist of only a handful of (subtokenized) words.

As can bee seen in the left figure of **??**, MLQA comprises quite a high number of examples — 868, to be precise — which do not fit subtokenized into a normal BERT model. However, those examples where the answer span lies inside the maximum BERT sequence are simply cut off after the maximum length and are kept in the data set for the experiments. Only the 22 exmaples where the answer span lies partially or fully outside the maximum length were dropped.

3.2.2.2 SOTA

Lewis et al. define two tasks tey use to evaluate performance on MLQA: The first one, cross-lingual transfer (XLT), means training models in English, and evaluating on the test sets for the various languages. The second case, generalized cross-lingual transfer (G-XLT), is not of interest to my thesis, since it involves a mixing of languages: teh context is presented in one language and the question in another. They include two models for the zero-shot transfer approach: multilingual BERT and XLM and use SQuAD (100,00 instances) as training set.

The best results for German in the XLT mode Lewis et al. reoprt, is a 47.6% exact match accuracy, achieved by XLM. However, as the training procedure and amount of traning data differs quite drastically frommy set up, it's not possible to directly compare their benchmark with my results. More on this in chapter 5.

3.2.3 PAWS-X

The PAWS-X corpus Yang et al. [2019] was compiled to provide a multilingual source for training models that address the problem of paraphrase identification. Since most corpora for this task are available only in English the authors compiled this corpus by humanly translate a subset of the original PAWS corpus Zhang et al. [2019a].

(3.16) Die Familie zog 1972 nach Camp Hill, wo er die Trinity High School in Harrisburg, Pennsylvania, besuchte.

1972 zog die Familie nach Camp Hill, wo er die Trinity High School in Harrisburg, Pennsylvania, besuchte.

True

(3.17) Prestige gehört der verheirateten Kiribati-Frau an, sie steht jedoch beträchtlich unter der Autorität ihres Mannes.

Die verheiratete Kiribati-Frau ist ein inhärentes Prestige, aber sie steht unter der Autorität ihres Mannes.

True

(3.18) Die österreichische Schule geht davon aus, dass die subjektive Entscheidung des Einzelnen, einschließlich des individuellen Wissens, der Zeit, der Erwartungen und anderer subjektiver Faktoren, alle wirtschaftlichen Phänomene verursacht.

Die österreichische Schule geht davon aus, dass die subjektive Entscheidung des Einzelnen, einschließlich des subjektiven Wissens, der Zeit, der Erwartung und anderer individueller Faktoren, alle wirtschaftlichen Phänomene verursacht.

False

(3.19) "Es ist der vierte Track und die dritte Single aus ihrem Durchbruch ""Smash"" (1994)."

Es ist der vierte Track und die dritte Single von ihrem Durchbruchalbum “ Smash ” (1994).

True

(3.20) Die Mannschaft reagierte auf die Änderungen im nächsten Spiel am selben Abend am 19. Februar.

Die Mannschaft reagierte auf die Änderungen im selben Spiel am nächsten Abend des 19. Februars.

False

blablabalblabdbasbdabsdbasbdlasdb[9]

(3.21) Die Single wurde am 12. Oktober 2012 im italienischen Radio Airplay gespielt und am 3. Dezember 2012 weltweit verschickt.

Die Single wurde am 12. Oktober 2012 nach Italien zu Radio Airplay geschickt und am 3. Dezember 2012 weltweit veröffentlicht.

(3.22) Lloyd gründete und leitete sein Unternehmen, um mit dem Verkauf von Spielzeug und Geschenken zu beginnen, und er erweiterte das House of Lloyd, mit Sitz in Grandview, Missouri, während das Geschäft mit den Geschenken wuchs.

Lloyd gründete und leitete sein Unternehmen zum Verkauf von Spiel- und Geschenkwaren und erweiterte das in Grandview, Missouri, liegende House of Lloyd mit dem Wachstum des Marktes für Geschenkwaren.

3.2.3.1 Preprocessing

During the preprocessing of this data set, the following considerations are taken into account:

In the predefined development and test splits, there are some examples where one or both sentences consist only of the string “NS”. I decided to not include this examples into the data used for training and evaluating my models, since those examples don’t contribute any useful features for the model.[10] Further, some examples consist of empty strings; I treat those the same way as the examples mentioned before.

[9] 3.21: False, 3.22: True

[10] The authors don’t comment on these obscure sentences, so I do not know what was the reasoning behind including these into the data sets.

Further, there are sentences XXXXX

3.2.3.2 Statistics

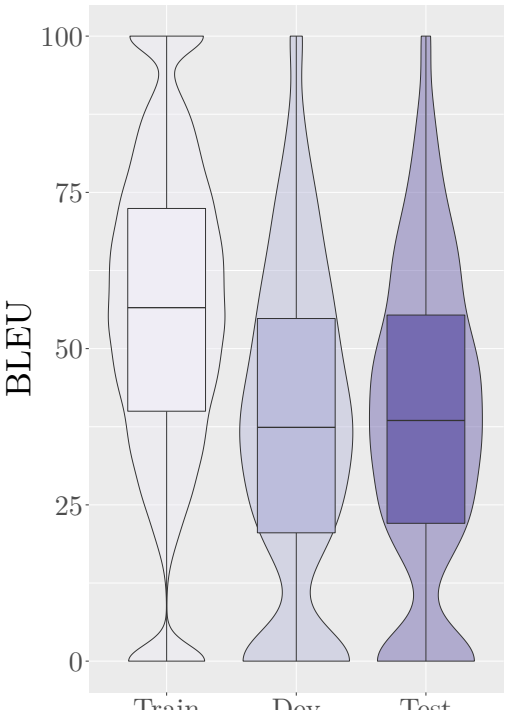
In figure **??** the lenght of the subtokenized sentence pairs are ploted. With the exception of one outlier in the first sentence group in the train set, the first and second sentences of all data sets seem to be of very similar lenghs. Taking into account the generation of the original PAWS data set, this is no surprise: "Our automatic generation method is based on two ideas. The first swaps words to generate a sentence pair with the same BOW, controlled by a language model. The second uses back translation to generate paraphrases with high BOW overlap but different word order." [Zhang et al., 2019a]

Since the training data are solely machine-translated while the development and test data are human-translated, there needs to be some clarification as to how differently those sets are. One measure to capture similarities between sentences is the BLEU score Papineni et al. [2002]: This score measures the overlap of n-grams between two sentences, such that XXX The BLEU score is a value between 0 (no n-gram overlaps) to 1 (perfect n-gram overlaps), where a BLEU score of 1 means that the two sentences are identical. As for other measures, like accuracy e.g., the value is sometimes multiplied by 100 for better readability, which I will also do here.

Mean BLEU-scores for sets:

Train: 55.27 stdev: 24.97
Development: 37.33 stdev: 25.57
Test: 38.37 stdev: 24.83

Er ist der derzeitige Weltmeister und Olympiasieger im Einzel und gilt mit einem Olympiasieg, drei WM- und zwei World-Cup-Titeln, sechs Siegen bei den World Tour Grand Finals und zahlreichen weiteren Titeln als einer der erfolgreichsten Tischtennisspieler überhaupt. Er ist Rechtshänder und verwendet als Schlägerhaltung den europäischen Shakehand-Stil. Er ist der derzeitige Weltmeister und Olympiasieger im Einzel und gilt mit einem Olympiasieg, drei WM- und zwei World-Cup-Titeln, sechs Siegen bei den World Tour Grand Finals



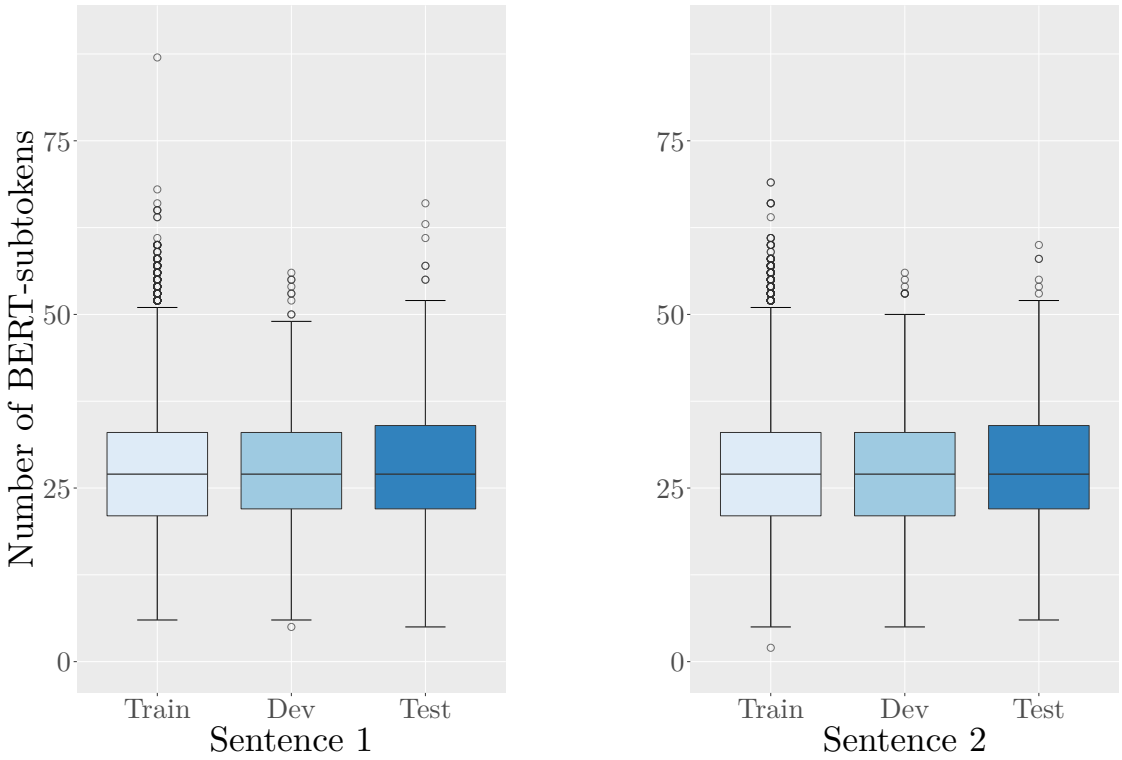


Figure 4: **Left**: Length of subtokenized PAWS-X first sentences. Note that one extreme outlier in the traning set comprising 863 (*sic*) BERT-subtokens is not included in the plot. **Right**: Length of subtokenized PAWS-X second sentences.

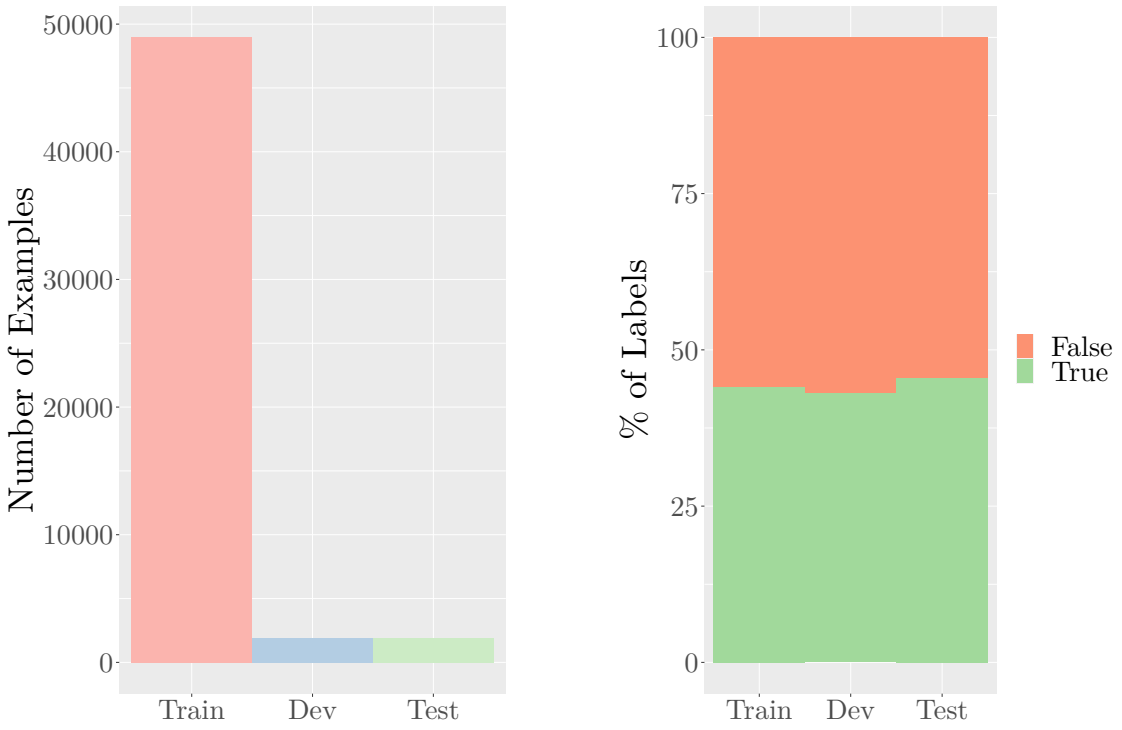


Figure 5: **Left**: Number of examples in the PAWS-X split. The automaticall translated trainin set contains way more examples than the humanly translated development and test set. **Right**: Distribution of paraphrase (True) and non-paraphrase (False) examples in the PAWS-X sets.

und zahlreichen weiteren Titeln als einer der erfolgreichsten Tischtennisspieler überhaupt. Er ist Rechtshänder und verwendet als Schlägerhaltung den europäischen Shakehand-Stil. Er ist der derzeitige Weltmeister und Olympiasieger im Einzel und gilt mit einem Olympiasieg, drei WM- und zwei World-Cup-Titeln, sechs Siegen bei den World Tour Grand Finals und zahlreichen weiteren Titeln als einer der erfolgreichsten Tischtennisspieler überhaupt. Er ist Rechtshänder und verwendet als Schlägerhaltung den europäischen Shakehand-Stil.

Number of instances:
Train: 48,977
Dev: 1,932
Test: 1,967

The BLEU scores indicate that the sentence pairs in the training set are in tendency much more similar to each other than in the development and test set. Taken into account how the data sets were generated, this makes actually sense, however: While the development and test sets were translated from English to German by humans, the huge training set was automatically translated. Since the original differences in the sentence pair might well have been rather subtle, it is no surprise that an algorithm might exhibit difficulties in grasping those differences; resulting in similar translations for two similar sentences. Note that due to the difficulties mentioned before, the automic translation resulted in 3,209 sentence pairs (6.6% of all the sentence pairs) with a BLEU score of 100.00 in the training set — which means they are identical.[11]

3.2.3.3 SOTA

Yang et al. [2019] achieve their best result — 89.2% accuracy for German — employing the following model architecture: They train a multilingual BERT on all languages, including the original English pairs and the machine-translated data in all other languages and evaluate on the individual languages.

[11] I reported this to the authors of the corpus, but didn't receive an answer from them.

3.2.4 SCARE

"Unlike product reviews of other domains, e.g. household appliances, consumer electronics or movies, application reviews offer a couple of peculiarities which deserve special treatment: The way in which users express their opinion in app reviews is shorter and more concise than in other product reviews. Moreover, due to the frequent use of colloquial words and a flexible use of grammar, app reviews can be considered to be more similiar [sic] to Twitter messages ("Tweets") than reviews of products from other domains or platforms [...]." [Sänger et al., 2016, p. 1114]

The Sentiment Corpus of App Reviews with Fine-grained Annotations in German Sänger et al. [2016] is a hand-annotated corpus that asserts so sentiment to German mobile app reviews stemming from the Google Play Store. Since there are many users of In contrast to other data sets, e.g. [Socher et al., 2013; Go et al., 2009], that attributes one sentiment label to a whole text (may it be a review, a tweet, etc.), Sänger et al. [2016] annotated their data set on a lower textual level: Not each review gets labelled for a certain polarity — i.e. *positive*, *negative*, or *neutral* — but what the authors call *aspects* and correlating *subjective phrases*. An aspect is an entity, that is related to the application: It may be the application itself, parts of the application, a feature request regarding the application, etc. A subjective phrase "express[es] opinions and statements of a personal evaluation regarding the app or a part of it, that are not based on (objective) facts but on individual opinions of the reviewers" [Sänger et al., 2016, p. 1116]. In other words, aspects are facts about the App and subjective phrases are user opinions regarding them. This fine level of annotations leads often to several annotations per review, the sentiment of which may not always match. As illustration, consider the following review:

(3.23) guter wecker... || vom prinzip her echt gut...aber grade was die sprachausgabe betrifft noch etwas buggy....[12]

There are the following annotations for the aspects and their corresponding subjective phrases (aspects are bold, the subjective phrase is italic and the polarity is normal):

- **Wecker**, *guter* → positive
- **Prinzip**, *echt gut* → positive
- **Sprachausgabe**, *etwas buggy* → negative

[12]The "||" denotes that the text left of it is the user given "title" of the review, and the part on the right is the actual review.

As is clear from this example, in a given review there may be several aspects with a corresponding subjective phrase per review. It is well possible, as in the provided example, that the sentiment of these is not always the same. The majority vote decision of the overall sentiment of the example above would be *Positive*.

(3.24) Ganz okay || Hatte ein Problem mit der APP aber die updates neu installiert und jetzt gehts wieder vorläufig mal Und Ordner wären schön wenn man diese erstellen kann

Neutral

(3.25) Ssssereeehhhr gut

Positive

(3.26) Wie kann man so eine gute app machen und dann nicht auf wvga anpassen. Weg mit den matschtexturen und vor allem dem Icon x-(

(3.27) spitze || Daran sollte sich MS ein Beispiel nehmen!

(3.28) Läuft nicht auf dem Acer A500 || Stürzt leider immer beim Abspielen eines Videos ab. Honeycomb 3.2

Example from .txt file:

ID	Text
7000	Alles wieder ok, das Update funktioniert wieder
7001	Echt super. || Schönes, und vor allem einzigartiges interface, wirklich klasse. Schön wäre noch, wenn man eigene lieder als klingeltöne einstellen könnte.
7002	Ein sicherer Start in den Tag
7003	timely wecker || Einfach nur top
7004	Super, aber || ändert klingeltonlautstärke. Nexus S Android 4.0.3
7005	Wecker || Wirklich gelungene app, tadellos!
7006	Sehr schöne UI und Optik... || Eine Bereicherung auf voller Länge... 5-Sterne ***** und daß gerne.
7007	Akkuverbrauch zu hoch || Wenn die app läuft dann ist der akku meines Note3 in ein paar stunden leer.
7008	NSA APP? || Innerhalb 2 Wochen 150MB an Hintergrunddaten?! Was wird da gesendet???
7009	Ist halt n Wecker

Table 3: An example from the alarm_clocks.txt file.

Example from .csv file:

Corresponding .rel file:

Class	ID	Left	Right	Text	Aspect- / Subj-ID	Polarity	Relation
subjective	7000	0	15	Alles wieder ok	7000-subjective2	Positive	Related
aspect	7000	21	27	Update	7000-aspect1	Neutral	Related
subjective	7000	28	40	funktioniert	7000-subjective1	Positive	Related
subjective	7001	0	10	Echt super	7001-subjective5	Positive	Related
subjective	7001	15	22	Schönes	7001-subjective4	Positive	Related
subjective	7001	38	51	einzigartiges	7001-subjective3	Positive	Related
aspect	7001	52	61	interface	7001-aspect2	Neutral	Related
subjective	7001	63	78	wirklich klasse	7001-subjective2	Positive	Related
subjective	7001	80	90	Schön wäre	7001-subjective1	Negative	Related
aspect	7001	113	135	lieder als klingeltöne	7001-aspect1	Neutral	Foreign

Table 4: An example from the alarm_clocks.csv file.

Relation-ID	Aspect-ID	Subj-ID	Aspect-String	Subj-String
7000	7000-aspect1	7000-subjective1	Update	funktioniert
7001	7001-aspect2	7001-subjective4	interface	Schönes
7001	7001-aspect2	7001-subjective3	interface	einzigartiges
7001	7001-aspect1	7001-subjective1	lieder als klingeltöne	Schön wäre

Table 5: An example from the alarm_clocks.rel file.

stats: there are 1,760 fine-grained annotated reviews

Baseline concerning imbalance labels: Always predicting majority class ("Positive") results in accuracy of 59.09%.

3.2.4.1 Preprocessing

For integrating the SCARE corpus into my GerBLUE corpus, I need to prepare the data, so it can be handled by the model architecture. Following the original GLUE sentiment task, the model needs only to predict one sentiment label for each example. Since there exist mostly multiple annotations for each review in this data set, the data needs to be pre-processed in a way, so that there is one review-label per example.

To generate the review-label, I simply carry out an majority class decision: The label that is most often annotated for a given review, regardless if it is an aspect or a subjective, is then also the review-label. If there is no majority label, the review-

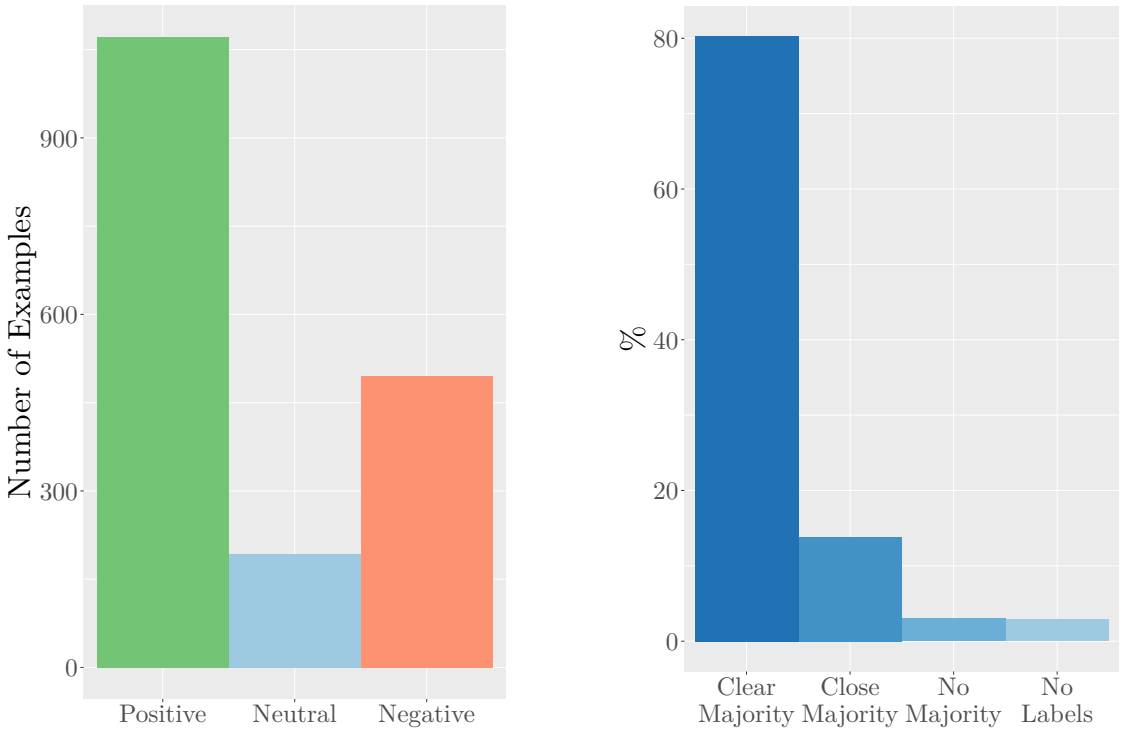


Figure 7: Left: Number of examples per label after heuristically computing them in the SCARE dataset. WRITE MORE ABOUT IMBALANCE, WHAT TO DO ABOUT IT, COMPUTE F1, ETC **Right**: Statistics of label generation. For most of the examples, there was a clear majority decision as to which label should be chosen. *Close Majority* means the majority vote was off by 1. The *No Majority*/*No Labels* portions in the graph were labelled *neutral* by default, while *Clear Majority*/*Close Majority* were labelled according to the majority vote decision.

label is set to "neutral". This is also the chosen strategy for 51 reviews that had no labels at all; an example of such a review is the following one:

(3.29) "Ich bin die erfuinderin || Ich bin die erfunden!!!!!!!!!!!!!!!!!!!!!!!!!!!!".

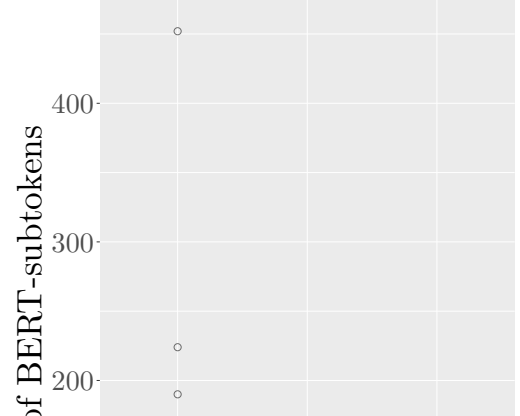
2.9% of reviews had no labels at all

3.0% of votes were non-majority

13.8% of votes were close (label difference of 1)

ARGUMENT FOR IMPLEMENTING AS ONE SENTENCE:

(3.30) Erkennt Drucker nicht... || ...,wenn er als Netzwerkdrucker an der FritzBox hängt, und eine manuelle Konfiguration über IP-Adresse ist nicht möglich.



3.2.4.2 Statistics

Number of examples:
Train: 1,232
Dev: 264
Test: 264

merged
average length train: 20.2 (sigma 21.6)

average length dev: 19.2 (sigma 19.1)

average length test: 20.6 (sigma 20.0)

subtokenized

average length train: 25.4 (sigma 28.2)

average length dev: 24.0 (sigma 23.1)

average length test: 26.1 (sigma 25.9)

3.2.4.3 SOTA

Sänger et al. [2016] don't predict a sentiment for each instance, but predict fine-grained aspect and subjective phrase spans using a CRF-based model. They report results for exact matches as well as partial matches. For the aspects, they achieve an F1 score of 69% and 80% for subjective phrases, respectively. Since predicting fine-grained aspect and subjective phrase spans is much more difficult than etrapolating an overall sentiment of the same utterance, a comparison between the outcomes of the two tasks are not really comparable. Furthermore,

3.2.5 XNLI

Conneau et al. [2018] built the XNLI corpus by employing professional translators to tranlsate 7,500 English sentence pairs from the Multi-Genre Natural Language Inference (MultiNLI) corpus Williams et al. [2017] into fifteen languages. First, they randomly sample 750 examples from each of the ten text source used in MultiNLI, which is in English, and then let the same MultiNLI worker pool generate three hypothesis for each sentence, one for each possible label (*entailment*, *contradicion*, *neutral*). Each sentence pair was then assigned a gold label that was retrieved by

carrying out a majority vote between the label that was assigned by the person who created the hypothesis and the labels that were assigend indepentently to the sentence pair by four other people. Finally, all the sentence pairs were translated into the different languages by tranlators. In addition, Conneau et al. [2018] carry out some tests to verify that the original gold label still holds in the translated sentences: They recruited two bilingual annotators to reevaluate 100 examples in English and French, i.e. they had to re-assign the labels given the sentence pairs. For tht English examples, the find a 85% concesus on the gold labels, and for French a corresponding 83%, from which they conclude that the overall semantic relationship between the two languages has been preserved.

(3.31) Ich wusste nicht was ich
vorhatte oder so, ich musste mich an einen
bestimmten Ort in Washington melden.

Ich war noch nie in Washington, deshalb
habe ich mich auf der Suche nach dem
Ort verirrt, als ich dahin entsandt wurde.

Neutral

(3.32) Natürlich haben
sie mich dort gefragt, warum ich ging.

Sie fragten, warum ich in den Laden ging.

Neutral

(3.33) Und ich dachte OK und das war es dann!

Nachdem ich ja gesagt hatte, endete es.

Entailment

(3.34) John Burke (Alabama) überprüft
und analysiert andere zeitgenössische
Konten und findet, dass Boswells
nicht nur der genaueste ist, sondern
er nutzt es, um Johnsons Charakter
zu demonstrieren, wobei andere es lediglich
als literarischen Geschwätz abstempeln.

John Burke ignoriert Aussagen.

Contradiction

(3.35) Die öffentliche
Bibliothek in Greenlee County, Arizona,
zeigt die finanziellen und technologischen
Probleme von ländlichen Einrichtungen auf.

Greenlee
County hat eine öffentliche Bibliothek.

Entailment

Here are the answers[13]

(3.36) Er kommt aus Griechenland
und er kommt aus einem kleinen Dorf in
Griechenland namens Tokaleka und er kam
nach Amerika und ich glaube es war 1969
oder 1970 und er heiratete kurz darauf.

Er ist ein griechischer
Mann, der kein Englisch spricht.

(3.37) Suchen Sie nach
Emily Dickinson's kommendem Gedicht,
alles was ich wirklich von Poesie wissen
muss habe ich bein Microsoft gelernt.

Dickinson schrieb Gedichte.

(3.38) „So
hat es auch in den kubanischen Tropen vor
Kurzem einen Tag gegeben, der so schön
wie Ruhm und kalt wie ein Grabstein war."

Es ist immer über 80 in Kuba.

3.2.5.1 Statistics

Number of Examples:
Train: 2,115
Dev: 374
Test: 5,009

[13] 3.36: Neutral, 3.37 Entailment, 3.38 Neutral

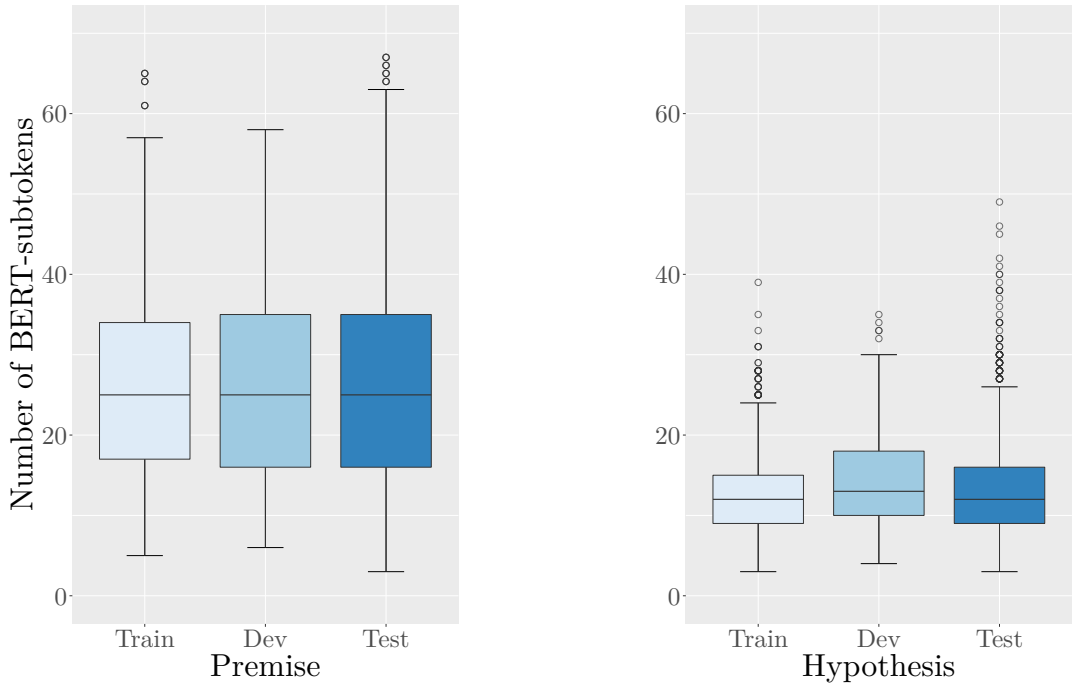


Figure 9: Left: Length of subtokenized XNLI premises. **Right**: Length of subtokenized XNLI hypotheses.

label distribution:

Neutral: 2,499 Entailment: 2,500 Contradiction: 2,499

In contrary to the above described PAWS-X corpus, there are no identical sentence pairs in XNLI.

3.2.5.2 SOTA

The best system Conneau et al. [2018] report for German on their XNLI data set is a model that relies heavily on translation: They train their BiLSTM on the MultiNLI data (432,702 instances) and translate the test set of the given language to English and predict on this data. Employing this startegy, the authors obtain an ac-

curacy on the German test set of 68.7%.

3.2.6 XQuAD

"XQuAD consists of a subset of 240 paragraphs and 1190 question-answer pairs from the development set of SQuAD v1.1 together with their translations into ten languages [...] In order to facilitate easy annotations of answer spans, we choose the most frequent answer for each question and mark its beginning and end in the context paragraph using special symbols, instructing translators to keep these symbols in the relevant positions in their translations" Artetxe et al. [2019].

(3.39) **Context**: Aristoteles lieferte eine philosophische Diskussion über das Konzept einer Kraft als integraler Bestandteil der aristotelischen Kosmologie. Nach Ansicht von Aristoteles enthält die irdische Sphäre vier Elemente, die an verschiedenen „natürlichen Orten" darin zur Ruhe kommen. Aristoteles glaubte, dass bewegungslose Objekte auf der Erde, die hauptsächlich aus den Elementen Erde und Wasser bestehen, an ihrem natürlichen Ort auf dem Boden liegen und dass sie so bleiben würden, wenn man sie in Ruhe lässt. Er unterschied zwischen der angeborenen Tendenz von Objekten, ihren „natürlichen

Ort“ zu finden (z. B. dass schwere
Körper fallen), was eine „natürliche
Bewegung“ darstellt und
unnatürlichen oder erzwungenen
Bewegungen, die den fortwährenden Einsatz einer Kraft erfordern. Diese Theorie, die auf der alltäglichen Erfahrung basiert, wie sich Objekte bewegen, wie z. B. die ständige Anwendung einer Kraft, die erforderlich ist, um einen Wagen in Bewegung zu halten, hatte konzeptionelle Schwierigkeiten, das Verhalten von Projektilen, wie beispielsweise den Flug von Pfeilen, zu erklären. Der Ort, an dem der Bogenschütze den Pfeil bewegt, liegt am Anfang des Fluges und während der Pfeil durch die Luft gleitet, wirkt keine erkennbare effiziente Ursache darauf ein. Aristoteles war sich dieses Problems bewusst und vermutete, dass die durch den Flugweg des Projektils verdrängte Luft das Projektil zu seinem Ziel trägt. Diese Erklärung erfordert ein Kontinuum wie Luft zur Veränderung des Ortes im Allgemeinen.

Question:
Wer leitete eine philosophische
Diskussion über Kraft?
Answer: Aristoteles

Question:
Wovon war das Konzept der
Kraft ein integraler Bestandteil?
Answer:
aristotelischen Kosmologie

Question:
Aus wie vielen Elementen
besteht die irdische Sphäre
nach Ansicht des Aristoteles?
Answer: vier

Question: Wo vermutete
Aristoteles den natürlichen
Ort für Erd- und Wasserelemente?
Answer: auf dem Boden

Question:
Was bezeichnete Aristoteles
als erzwungene Bewegung?

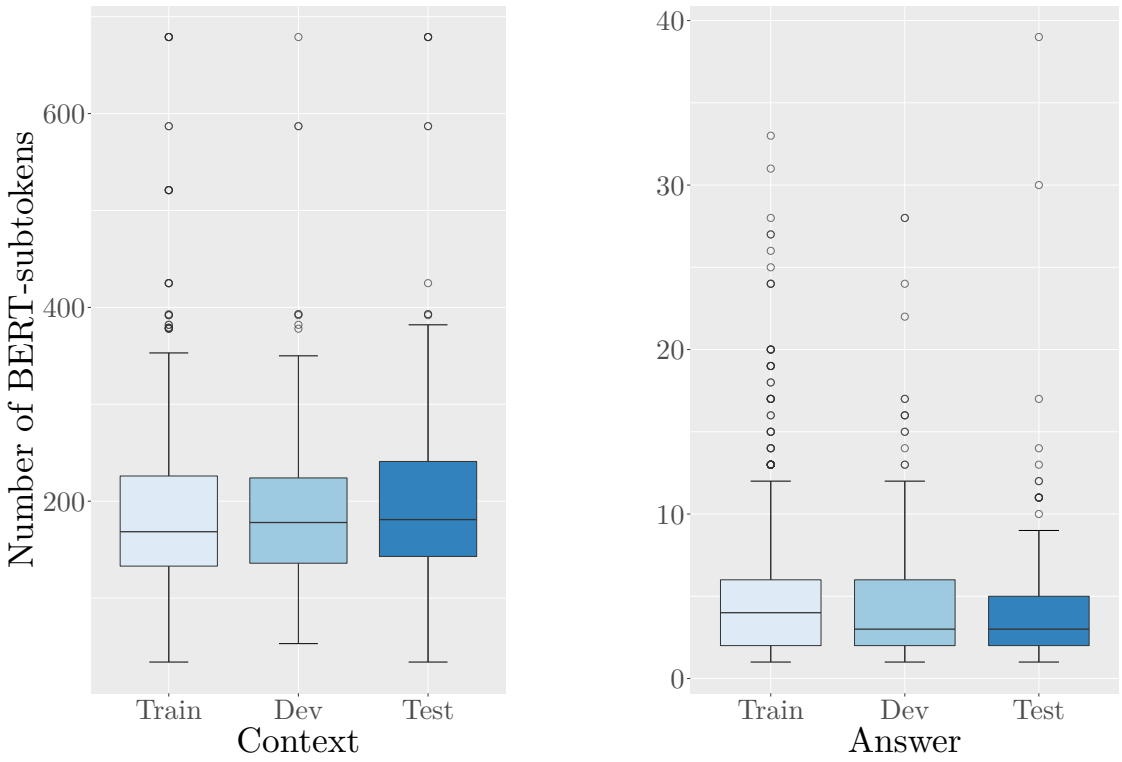


Figure 11: Left: Length of subtokenized XQuAD contexts. **Right**: Length of subtokenized XQuAD answers. Note the difference in y-axis scaling between the two plots.

Answer: unnatürlichen

Artetxe et al. [2019]

3.2.6.1 Statistics

Nxamples:
Train: 820 Dev: 181 Test: 178

(3.40) **Context**: Da sowohl Präsident Kenyatta als auch Vizepräsident William Ruto 2013 Gerichtstermine vor dem Internationalen Strafgerichtshof in Verbindung mit den Auswirkungen der Wahl von 2007 hatten, entschied sich US-Präsident Barack Obama, das Land während seiner Afrikareise Mitte 2013 nicht zu besuchen.

Später in diesem Sommer
besuchte Kenyatta auf Einladung
von Präsident Xi Jinping China,
nachdem er einen Zwischenstopp
in Russland eingelegt und die USA
als Präsident nicht besucht hatte.
Im Juli 2015 besuchte Obama
Kenia als erster amerikanischer
Präsident, der das Land während
seiner Regierungszeit bereiste.

Question:
Welches Land besuchte Kenyatta
auf Einladung des Präsidenten?[14]

Question: Wann besuchte
Obama Kenia schließlich?[15]

Question: Wer entschied sich, das
Land 2013 nicht zu besuchen?[16]

Question: Was war
das Ergebnis der Wahl von 2007?[17]

3.2.6.2 SOTA

Very peculiar architecture that consits in re-training a monolingual English BERT model on Wikipedia and transfer it to target language following these steps:

1. Pre-train a monolingual BERT in English with original pretraining objectives

[14] China
[15] Juli 2015
[16] US-Präsident Barack Obama
[17] Gerichtstermine vor dem Internationalen Strafgerichtshof

2. Transfer model to new language L_2, but learn only token embeddings new (transformer body is frozen) with original pretraining objectives

3. Fine-tune transformer for downstream task in English (transformer body is freezed)

4. Zero-shot transfer this model to L_2 by swapping the English token embeddings with the L_2 embeddings

   The authors report the following results for the German part of XQuAD: F1: 73.6 Accuracy (exact match): 57.6%

3.2.7 Summary

GerGLUE

		NLP Task	ML Task	# Examples Train/Dev/Test	Predefined Splits	Register coll. < mix. < form.	Remarks
		Classification					
Single	deISEAR	Emotion Detection	Multi-Class	1001/150/151	-	mixed	Boilerplate text structures ("Ich fühlte [?], als ...")
Single	SCARE	Sentiment Analysis	Multi-Class	1,232/264/264	-	colloquial	Very informal, ungrammatical, and often short text snippets
Pair	PAWS-X	Paraphrase Identification	Binary	48,977/1,932/1,967	Train/Dev/Test	formal	Translation artifact noise, dev/test splits OOD to train
Pair	XNLI	Natural Language Inference	Multi-Class	2,115/374/5,009	Dev/Test	mixed	Translation artifact noise, language from different domains
		Question Answering					
Pair	MLQA	Question Answering	Span Prediction	432/77/4,499	Dev/Test	formal	Highly imbalanced splits regarding number of examples
Pair	XQuAD	Question Answering	Span Prediction	820/181/178	-	formal	Small data set

Table 6: Overview of *GerGLUE* data sets and tasks. The number of exmaples represent the size of the set splits after preparing the data sets for the experiment; therefore all datasets have all splits. Note that during the preprocessing some examples had to be excluded (see chapter 3 for more details), that's why some numbers do not coincide with those of table 2.

4 Architecture

4.1 Overview

GliBERT is an architecture that combines different, pre-existing models and tools. The general way an input sequence is processed by GliBERT is depicted in figure 12:

"Throughout this work, a «sentence» can be an arbitrary span of contiguous text, rather than an actual linguistic sentence. A «sequence» refers to the input token sequence to BERT, which may be a single sentence or two sentences packed together." [Devlin et al., 2018]

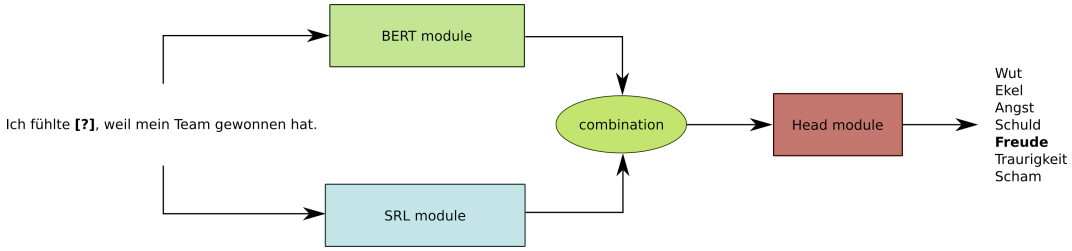


Figure 12: General architecture of GliBERT, exemplified for the deISEAR task.

The core parts of the model are the following:

BERT module This is the vanilla BERT base model: It tokenizes the input sequence and sends it through its twelve transformer layers and outputs the final hidden states of each (sub-)token.

SRL module This modules actually consists of two submodules: First, the sequence is processed by ParZu [Sennrich et al., 2009] to identify predicates. Second, the sequence with the information about which tokens are predicates is handed to the DAMESRL model [Do et al., 2018] which predicts actual SRLs. To ensure there are no tokenization mix-ups between BERT and DAMESRL (because these differences are not reversible as will be seen later), the sequence gets tokenized BERT-style and is passed as a list of tokens to DAMESRL.

combination To combine the BERT embeddings and SRLs, first the SRLs are transformed into numerical representations, or, in other words, are embedded into a comparably lo-dimensional space. secondly, the BERT and SRL embeddings need to be processed, i.e. splittet or merged, respectively, so that they can be concatenated. For this, there exist two approaches: (A) Fuse the subtokens of BERT back to tokens, (B) Split the SRLs according to the subtokens of BERT.

Head module At last, the combined representation of the input is fed through the final network that transforms it to predict task-dependent output. Several architectures can be applied here: FFNNs, GRUs, CNNs, etc.

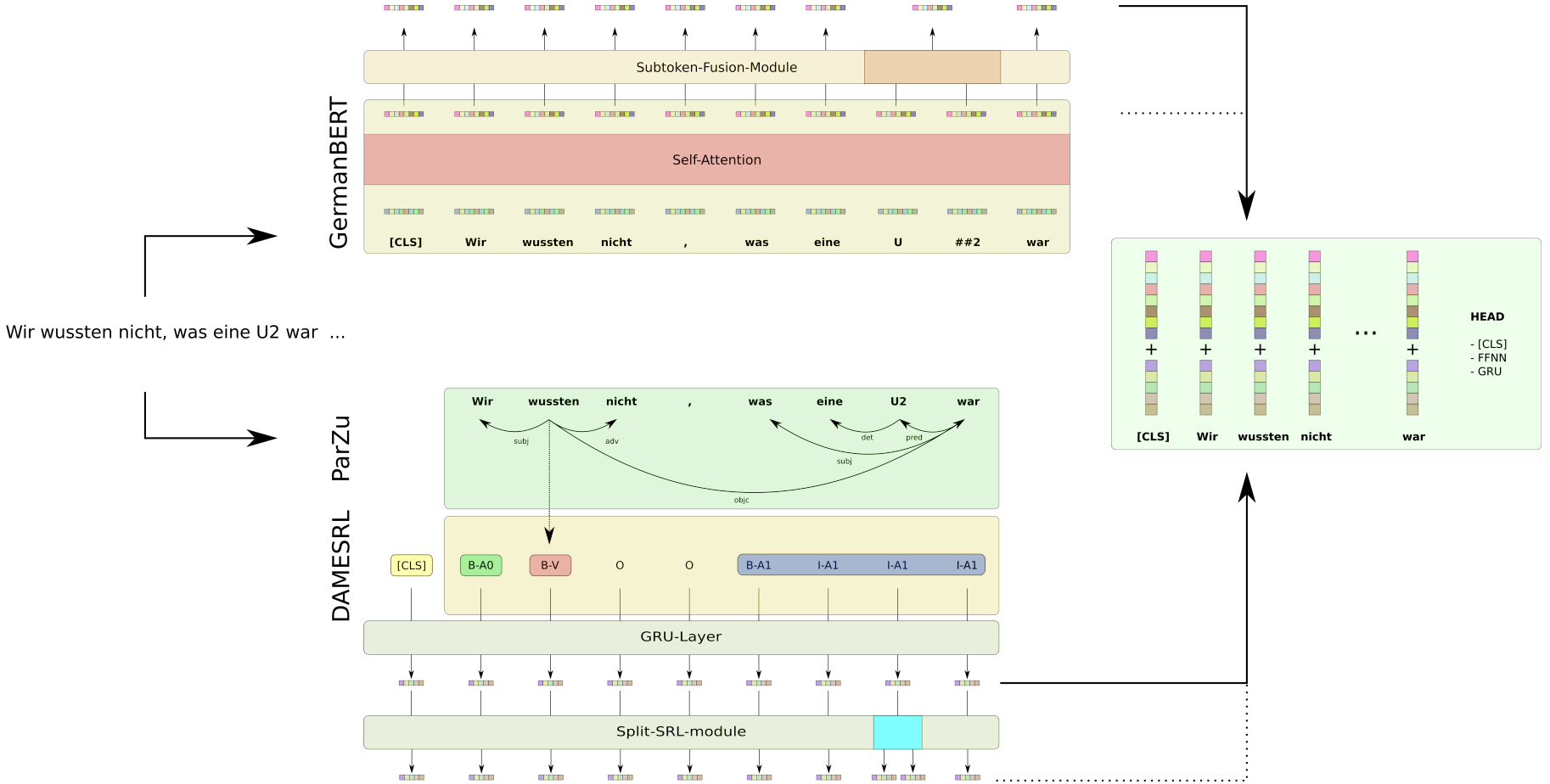


Figure 13: Detailed architecture of GliBERT: On the left, the input sentence is passed through two paths: On top, through the German BERT, with the optional subtoken fusion module on top. On the bottom, through ParZu and DAMESRL, with subsequent embedding via a GRU model; after that an optional split token module follows. The bold arrows on the right side show information flow, when BERT subtokens are fusioned for appending with SRLs. The dotted arrows represent the information flow when the SRLs are splitted to match with the BERT subtokens.

4.2 BERT module

Since its publishing two years ago, BERT [Devlin et al., 2018] has often been viewed as a "turning-point" in NLP: The embeddings it computed only by implementing massive "unsophisticated" self-supervised pretraining proved to be very potent representations of language and were successfully implemented in a wide array of downstream tasks. Pretrained models and APIs for BERT are vastly available — I chose to use the `bert-base-german-cased` model from deepset which is available in pyTorch through the hugging face libraryWolf et al. [2019].

deepset only provides a BERT base model which has the following specifications according to it's configuration file:

- Transformer Blocks: 12
- hidden Size: 768
- hidden activation function: GeLu
- hidden dropout probability: 0.1
- Attention Heads: 12
- Vocabulary size: 30,000 (cased)
- Total Parameters: 110 million

The handling of the BERT model is straightforward through huggingface's `transformer` library: With a simple function call `BertModel.from_pretrained()` one loads the pretrained BERT, and with another function, `BertTokenizer.from_pretrained()`, one instantiates the BERT tokenizer. After encoding a sentence using the tokenizer's method `.encode_plus()`, the encoded sentence is send through BERT via it's `.forward()`-method — or called implicitly, by simply passing the sentence to the model — which returns the vectors for all input tokens, which can then be used in downstream tasks. Fine-tuning is simply done by passing the computed loss to the specified loss funtion (I use the AdamW loss function [Loshchilov and Hutter, 2019], a modification of the well-known Adam (Adaptive Moment Estimation) loss function [Kingma and Ba, 2014], implementing weight decay), which updates the BERT weight matrices.

In the GitHub repository, in the file `load_data.py`, the data gets tokenized and loaded, and in the file `gli_bert.py`, the forward pass and weight-updating is defined in the `fine_tune_BERT()` function.

4.3 SRL Module

A Semantic Role Labeller (SRL) is a system, that assigns automatically semantic roles to a given input text.[1]

State-of-the-art semantic role labellers (SRLs) are end-to-end models, nowadays often implementing deep learning techniques, like RNNs or self-attention mechanisms, that render tedious feature engineering unnecessary. For my system, I implement the DAMESRL, a model presented by Do et al. [2018]. I use their pre-trained German Character-Attention model which, according to the authors, achieved an F1 score of 73.5% on the CoNLL'09 task [Hajič et al., 2009]. However, their SRL needs as input not only the sentence, but also "its predicate w_p as input" [Do et al., 2018].

"A major advantage of dependency grammars is their ability to deal with languages that are morphologically rich and have a relatively free word order." [Jurafsky and Martin, 2019, p. 274] For extracting predicates, I rely on the dependency tree the ParZu parser Sennrich et al. [2013] generates for a given sentence. Given the parsed sentence, I have to decide what tokens in it are predicates, and which are not. While this may seem like a straightforward task — just find the verb as in a simple sentence like "He *ate* the apple." —, there are actually a few caveats (predicates are emphasised): (1) There may be no predicates at all: "What a day!". (2) There might be more than one predicate: "We *saw* her *leave* the room". (3) Not all verbs might be predicates: "I can *hear* you". In the following section, I will describe how I tackle these problems by making use of parsing information from ParZu.

4.3.1 Finding Predicates

It is a known problem in the analysis of semantic roles that a proper procedure for predicate identification is a problem hard to tackle, consider e.g. the discussion concerning so called light verbs: Wittenberg [2016].

"First, the predicates which assign semantic roles to the constituents are identified prior to semantic role labelling proper. They are usually identified as the main verbs which head clauses." [Samardzic, 2013, p. 74] In a dependency framework like the Universal Stanford Dependencies (USD) [De Marneffe et al., 2014], which explicitly sets the content verb as root, identification of the relevant predicate is straightforward: One has simply to look at the dependency parse tree of a given

[1] This may be one or multiple sentences.

sentence and select the verbal heads — i.e. roots — of the clauses. However, the ParZu parser models not content verbs as heads but function verbs.[2]

(interestingly, this stands in contrast to the Pro3Gres parser [Schneider, 2008] which

"In a constituency parse, the finite verb is the head of a verb phrase or rather sentence. A dependency parse, on the other hand, does not consider auxiliaries as heads and therefore finite verbs are usually not the head of the sentence. Hence, the head of a sentence typically is the verb containing the meaning. In that sense, dependency structures are closer to the semantics of a sentence." [Aepli, 2018, p. 6f.]

According to the USD, function words are subordinated to content words, which means that in a sentence "He was hit by a ball.", the infinite participle *hit* would be analysed as root, not the finitely inflected *was*. This is an accordance with the view that XXXXXXXXXX However, there is a "substantial amount of evidence [that] delivers a strong argument for the [...] approach, which subordinates full verbs to auxiliaries" Groß and Osborne [2015].

"The parsing scheme that USD advocates takes the division between function word and content word as its guiding principle. One major difficulty with doing this is that the dividing line between function word and content word is often not clear." Groß and Osborne [2015]

Following Foth [2006]

(4.1) Die Keita-Dynastie regierte das vorkaiserliche und kaiserliche Mali vom 12. Jahrhundert bis Anfang des 17. Jahrhunderts.

(4.2) Im tibetischen Buddhismus werden die Dharma-Lehrer/innen gewöhnlich als Lama bezeichnet.

(4.3) Die Klage wurde abgewiesen, was als Sieg beschrieben werden kann.

whose dependency parse tree is shown in Figure 14: This sentence has five verbs in it, *wurde*, *abgewiesen*, *beschrieben*, *werden*, and *kann* (POS-tag "V" in the second row), but only two of them are relevant predicates, i.e. predicates that carry "true" semantics.

I propose the following algorithm 1 deciding whether a verb in a sentence is or is not a predicate using a heuristic, relying on the token's POS tag that the parser

[2]This follows general dependency frameworks proposed for German, e.g. Gerdes and Kahane [2001]; Groß and Osborne [2015].

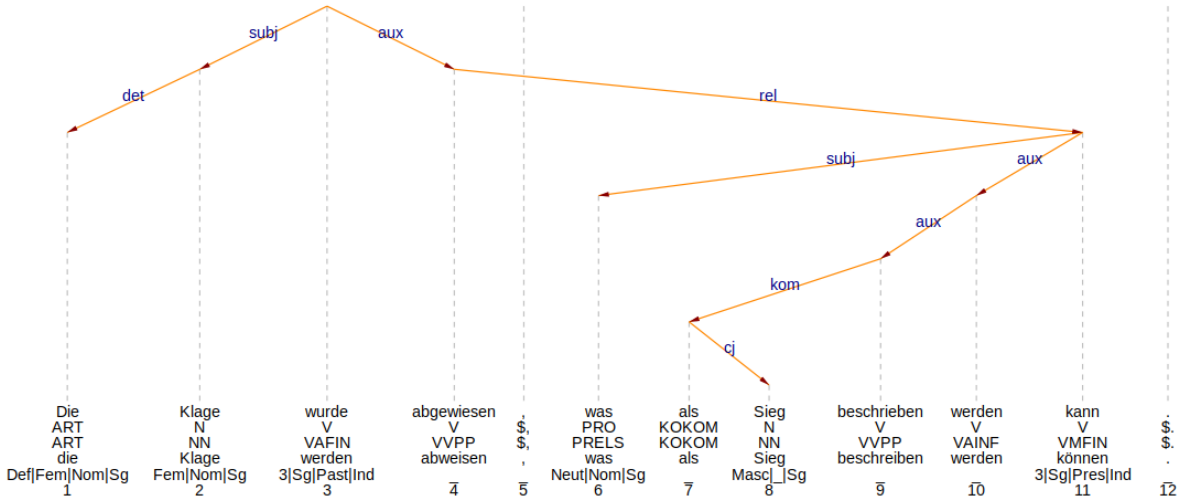


Figure 14: Example dependency parse tree for a sentence with multiple predicates.

predicts. The ParZu parser's default output follows the CoNLL scheme [Buchholz and Marsi, 2006] which means that there are two levels of POS tagging: coarse-grained (CPOSTAG) and fine-grained (POSTAG), where the POSTAG corresponds to the token's STTS tag [Schiller et al., 1999].

The condition on line 9, that only tokens in the respective subclause are considered, is ensured by making sure that if a token u's POS is "V" and it points to its head t, that it is not itself the head of a subclause — i.e. its dependency relation is e.g. "relative clause". If that is the case the token u is considered to belong to another subclause and therefore not preventing token t from getting labelled as a predicate. Consider again the example 4.3.1: Let's say we are in the for-loop at the token *weitergeleitet*. Because it is a verb but not a finite full-verb, we enter the else-clause on line 7. If we were now to loop through all token of sentence 4.3.1 we would find that token *führt* is a verb that points to our primary token. Without the above outlined constraint that only verbs in the same subclause pointing to our original verb are preventing it from being labelled a predicate, *weitergeleitet* would be labelled as non-predicate. This is obviously false. Taking into account the above considerations, we see that although *führt* points to *weitergeleitet*, its edge label is *rel* — which means that it's the head of a relative subclause — therefore it is not anymore in the same subclause and *weitergeleitet* gets labelled as predicate.

Algorithm 1 Predicate finding algorithm

```
1:  for all token t ∈ sentence do
2:    if CPOSTAG t ≠ 'V' then
3:      t ← NOT_PRED
4:    else
5:      if POSTAG t = 'VVFIN' then
6:        t ← PRED
7:      else
8:        FLAG ← True
9:        for all token u ≠ t ∈ subclause where t ∈ subclause do
10:         if CPOSTAG u = 'V' ∧ u dependent on t then
11:           t ← NOT_PRED
12:           FLAG ← False
13:           break
14:         end if
15:       end for
16:       if FLAG = True then
17:         t ← PRED
18:       end if
19:     end if
20:   end if
21: end for
```

4.3.2 Ensuring Tokenization Equivalence

One of the major difficulties I ran into, was the tokenization differences between different parsers. In conrete terms, this means that it is not always possible to correctly align the tokens which two parsers produce for the same sequence. The DAMESRL system implements the tokenizer provided by the Natural Language Toolkit (NLTK)[3] which implements a linguistically motivated tokenizing. explain what that means BERT, in contrast, utilizes an approach called "WordPieces", which is a rather information processing motivated approach: "Using wordpieces gives a good balance between the flexibility of single characters and the efficiency of full words for decoding, and also sidesteps the need for special treatment of unknown words." [Wu et al., 2016, p. 2]. Although the NLTK algorithm is guided by linguistically informed rules and statistic while the WordPieces approach simply reflects distribution properties of assembled letters, both systems tokenize sentences in the same way in most cases. However, especially when rare symbols such as currencies, units, and the like are present, the tokenization slightly differs What is even worse,

[3] `https://www.nltk.org/`

often the correct alignment of those differing sequences is rather complicated to obtain. automatically

To illustrate this, consider the following sentence from the PAWS-X data set:

(4.4) Die mittlere Oberflächentemperatur wird auf -222 °C (~51 K) geschätzt.

BERT (merged)	**NLTK**
Die	Die
mittlere	mittlere
Oberflächentemperatur	Oberflächentemperatur
wird	wird
auf	auf
-	-222
222	°C
[UNK]	(
(	~51
~	K
51	)
K	geschätzt
)	.
geschätzt	
.	

The first question that arises is: which tokenization should be mapped onto wich? In other words: should we try to align the BERT tokens with the corresponding NLTK tokens or vice versa? Let's assume we decide to align the tokenization T with fewer items to the one with more items — in this case this would mean aligning T_{NLTK} with T_{BERT}. So, the first five tokens are no problem, we can align them by simply doing an exact match and confirm that the elements correspond. But when we reach the sixth token, the exact match fails. To decide whether the token $t_{T_{BERT}}$ or the token $t_{T_{NLTK}}$ was split up — i.e. to determine which token must be copied to ensure tokenization equality —, we need to do a mutual substring match. Doing this, we eoudl find out that "-" is a substring of "-222". In consequence, we align the two, duplicate "-222" and compare it with token number 7 in T_{BERT}. Since "222" is a substring of "-222", so we align the two of them.

While it is theoretically possible to align tokens that were differently tokenized by the two algorithms, it is nevertheless quite cumbersome. The main problem, however, arises due to the [UNK] token BERT introduces for characters — or character sequences — which lie out of its vocabulary. Since there is obviously no more (sub-

)string comparison possible, the process gets even more complicated: Suppose you have duplicted the "-222" in the NLTK column and are now on line 7. In the BERT tokenization you see the "[UNK]" token, while in the NLTK you see a "°C". To find out, what all is containt in the "[UNK]", you need to look at the token before and after it in the BERT tokenization and compare it with the respective NLTK tokens. since the the and so on.... the BERT-tokenized sequences, to get around this issue.

BERT (merged)	**NLTK**
Die	Die
mittlere	mittlere
Oberflächentemperatur	Oberflächentemperatur
wird	wird
auf	auf
-	-222
222	-222
[UNK]	°C
(	(
~	~51
51	~51
K	K
)	)
geschätzt	geschätzt
.	.

4.3.3 DAMESRL

There are not too many SOTA SRL frameworks available for German that come with a pre-trained model, especially such ones that can be integrated in a pipeline in a pipeline of a bigger system.

Do et al. [2018] fill exactly this hole: They introduce DAMESRL, an SRL framework that implements SOTA architecture, namely self-attention mechanisms, similar to BERT's. They report an F1 score of 73.5 for their best model configuration on the German data set of CoNLL '09. This best configuration is based on word as well as character embeddings, self-attention and a softmax layer on top.

The DAMSRL predictor receives the BERT-tokenized sentence along with the information which tokens in it are predicates (zero or more). For each token labelled as predicate in a sequence it predicts for each other token in the sequence its SRL.

4.3.4 GRU

Finally, the predicted SRLs need to be encoded in a numeric way so they can be concatenated to the vectors which are computed by BERT. The "classic", pre-transformer age times, way of encoding sequential data would be to employ a recurrent neural network architecture. I decided to use GRUs, since they are less computational intesive and research has found both architectures mostly performing on par (cf. Chung et al. [2014]).

In most cases, a sentences contains not exactly one predicate which distributes semantic roles, but several, especially in longer sentences, — or even none, especially in colloquial, short sentences. Research by Zhang et al. suggests that fixing the number of predicate-argument structures to three yields the best results; so I adopt this number. In other words, if a sentence has more than three argument-predicate structures, I only care about the first three predicates identified (if proceeding from left to right through the sequence), and disregard the others. However, if there are fewer than three predicate-argument structures present, I test and report results for two strategies: The first solution (the left sentence in SRL 4.1) lies in filling the "unfilled" predicate-slots with the special "0"-SRL. The second (the right sentence in SRL 4.1) simply copies the first predicate-argument structure to the unfilled slots, thus amplifying the signal from the first predicate-argument structure.

	Slot 1	Slot 2	Slot 3		Slot 1	Slot 2	Slot 3
Wir	B-A0	0	0	Wir	B-A0	B-A0	B-A0
wollten	0	0	0	wollten	0	0	0
eine	B-A1	0	0	eine	B-A1	B-A1	B-A1
Sache	I-A1	0	0	Sache	I-A1	I-A1	I-A1
mehr	I-A1	0	0	mehr	I-A1	I-A1	I-A1
retten	B-V	0	0	retten	B-V	B-V	B-V
als	B-C-A1	0	0	als	B-C-A1	B-C-A1	B-C-A1
die	I-C-A1	0	0	die	I-C-A1	I-C-A1	I-C-A1
Restlichen	I-C-A1	0	0	Restlichen	I-C-A1	I-C-A1	I-C-A1
.	0	0	0	.	0	0	0

SRL 4.1: The two strategies for dealing with less than three predicates: **Left**: The open SRL slots get filled with the special SRL *0*. **Right**: The first SRL structure gets duplicated until all slots are filled.

- number of predicates equals 3
- duplicate or zeros when not enough predicates
- concatenate all sentences and then encode vs. encode each sentence and then concat
- ...

- Embed each sentence alone vs. concatenate all sentences
- same for text_1, text_2
- add meta-SRLs [CLS], [SEP]
- duplicate if less preds than 3 vs. fill with zeros

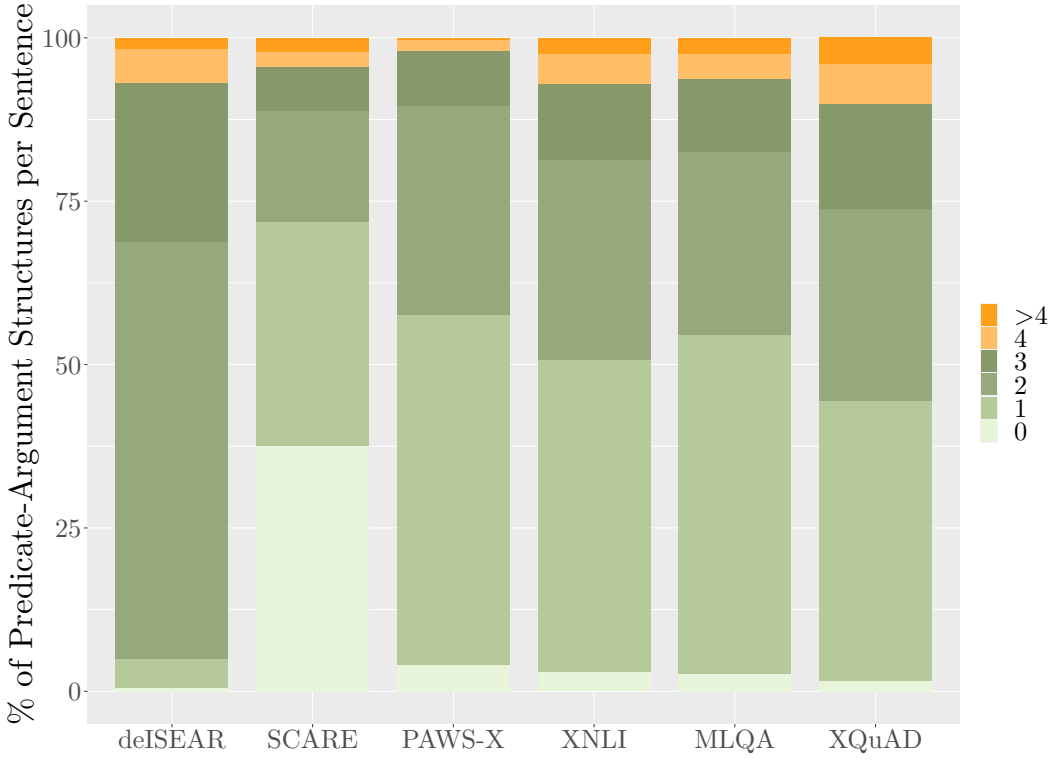


Figure 15: Number of predicate-argument structures in all data sets. Due to it's boilerplate template form, deISEAR shows a peculiar distribution: Since it's examples always begin with "Ich [$_{\text{PRED}}$ fühlte] X, als ....", it's guaranteed that at least one predicate is identifed, and it is very probable, because of this sentence structure, that another will occur. The other curious pattern exhibits SCARE: In no other data set the amount of sentences where ParZu couldn't detect any predicates is that high.
Besides the noted peculiarities, it is safe to say that setting the maximum number of predicate-argument structures to three does probably not lead to much information loss; on average over all datasets, 92.73% of all sentences have three or less such structures.

4.4 combination

4.4.1 Aligning BERT subtokens with SRL tokens

A crucial part in the overall architecture is the combining of the numeric representation of (sub-)words computed by the BERT network and the embedded SRLs.

One difficulty lies in the fact that, as already mentioned above, BERT has its own tokenizer which implements a so-called sub-word or wordpiece Wu et al. [2016] encoding strategy: BERT has a fixed length of vocabulary it can hold, namely 30,000 tokens. The wordpiece tokenization approach is a balance between word and character tokenization in that that it "gives a good balance between the flexibility of single characters and the efficiency of full words for decoding, and also sidesteps the need for special treatment of unknown words" [Wu et al., 2016, p. 2]. As a consequence, BERT tokenizes a sentence quite differently than a traditional parser, since the latter adheres to the full tokens. Consider the following example:

(4.5) Es ist der Sitz des Bezirks Zerendi in der Region Akmola.

(4.6) ParZu: Es ist der Sitz des Bezirks Zerendi in der Region Akmola .

(4.7) GermanBERT: Es ist der Sitz des Bezirks Zer ##end ##i in der Region Ak ##mol ##a .

A further challenge besides the alignment of traditional tokenization and wordpiece tokenization is the general difference in parsing a sentence that exist.

4.5 Head Module

4.5.1 Classifiction

While for question answering there was little tweeking needed to adapt to the extended BERT embeddings, for classification the situation looks a bit more complex. The standard BERT way of doing classification tasks runs as follows:

- Prepare the data: add a [CLS] token at the beginning, a [SEP] token between the two sentences (if there are), and pad with the [PAD] token
- Send the prepared examples through the BERT network
- Select only the embedding for the first token — i.e. the [CLS] —, send it through a dense layer with a softmax and predict the class for this example

Devlin et al. [2018] visualize this as can be seen in figure 16.

The problem now is that in the above described standard implementation, there is no straigthforward way to enrich the BERT embeddings with SRLs, since the only embedding that is used for prediction is the [CLS] token; since this is a special BERT token it is not present in the original sentence and, therefore, it does not have

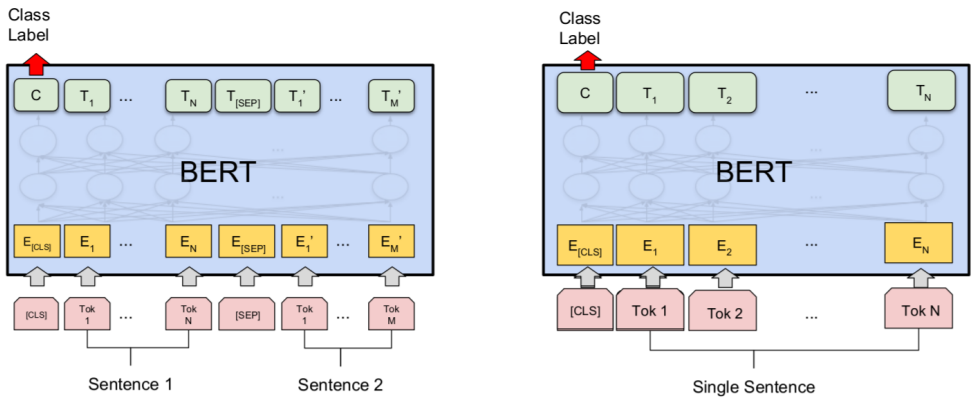


Figure 16: Schema for sentence pair (left) and single sentence (right) classification. Figure taken from [Devlin et al., 2018].

a corresponding SRL. (And what should that be? Since it is a meta-token it rcan't really have a SRL?

To tackle this problem, I remodeled the architecture from Zhang et al. [2019b] on the one hand, and on the other hand tested several other final layer designs.

4.5.1.1 [CLS] Head

In their paper for SemBERT, Zhang et al. [2019b] do not really address the issue laid out above. To the contrary, the differnt pieces of information they provide are rather conflicting, only after inspecting the code they released on GitHub, the picture somewhat cleared:

After predicitng the SRLs for a given input, they add pseudo-SRLs for the [CLS] and [SEP] tokens. In the look-up table of the BiGRU that consumes the SRLs, they then simply add the corresponding keys — so that besides regular SRLs as "B-V" (beginning of predicate) or "I-A0" (inside or end of argument zero), there are also the labels "[CLS]" and "[PRED]" After sending this sequence through the BiGRU, they concatenate the two hidden states of the [CLS] SRL with the [CLS] BERT embedding and predict on that vector SRL for the [CLS] token after the sequence was sent through the BiGRU which then can be appended to its BERT embedding vector

Formally, the [CLS] Head is a one layer Fullyconnected Feedforwrd Neural Network. The prediction is the results of sending the embedding of the [CLS] token through the network: $\hat{y} = softmax(L_1)$. The input dimensions of the layer vary according to

the setting, i. e. if the SRLs are included, or not. if not, then the demsion is simply the standard BERT hidden size $h_i \in \mathbb{R}^7 68$, if the SRL embedding is appended, then $h_i \in \mathbb{R}^7 88$. the output dimensionality of the network depends on the number of classes for the task: This number ranges from 2 (PAWS-X, *true*, *false*) to 7 (deISEAR, *Traurigkeit, Scham, Freude, Angst, Wut, Ekel, Schuld*).

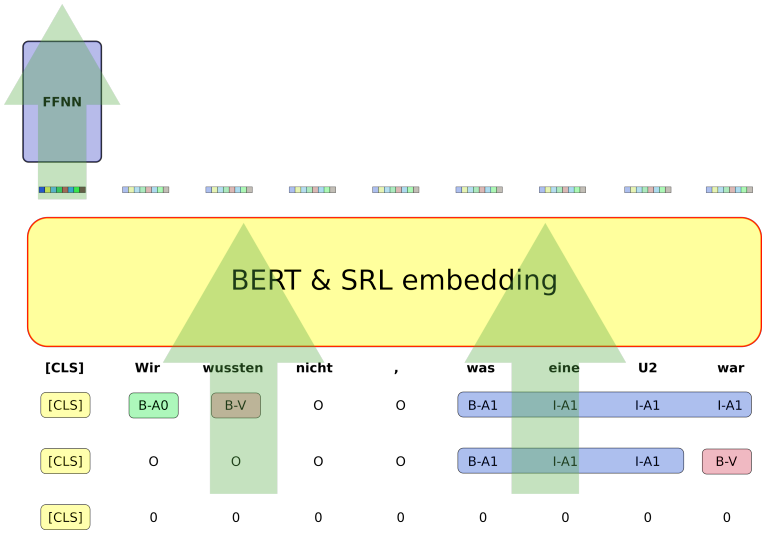


Figure 17: Head with a one-layer **F**eed **F**orward **N**eural **N**etwork on the [CLS]-token

In table XYZ I report the gains, losses this strategy yields for the four classifications data sets in my corpus.

4.5.1.2 FFNN Head

FFNN stands for Feed Forward Neural Network. While th [CLS] Head procudes predictions based on the weights of the last layer output of the [CLS]-token, this head takes the last layer output of all tokens and concatenates them. While the approach implemented by Zhang et al. [2019b] is able to improve the vanilla BERT approach, it does not lead to an improvement on others, or worse, brings the perormance down. I suspect a reason for this may lie in the manner of how the SRLs are processed in this approach. The information SRLs provide is, what may be called, sub-sentence specific and cannot be adequately represented as a single information piece. By this I mean that it does not suffice to know that given an utterance x that there is a specific SR in it; rather the information *where* is crucial. Consider the following example (the pseudo SRL [CLS] is added):

$[_{\text{[CLS]}}\ ]\ [_{\text{A-0}}$ The man$]\ [_{\text{predicate}}$ asked$]\ [_{\text{A-1}}$ his friend$]$.

After the subscripted SRLs were consumed by the BiGRU, there is some information about all SLRs in the hidden states of the [CLS] token. While there may be some

information about there being a predicate, an argument zero, and an argument 1 present, it is completely impossibly to determine from which tokens these signals came. Especially in sentence pair tasks, such as paraphrase identification, this information is however absolutely crucial. As can be seen from results 7, this hypothesis is also supported by the results:

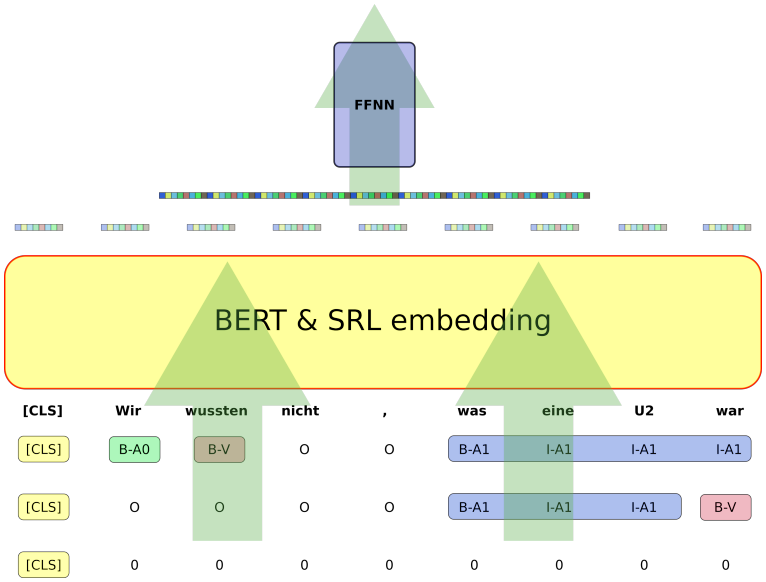


Figure 18: Head with a one-layer **F**eed **F**orward **N**eural **N**etwork on the concatenated token sequence.

As has been shown by e.g. Myagmar et al. [2019] for sentiment analysis, a simply final fully-connected feed forward layer produces fairly good results (in fact, it performs the best in the different architectures they tested for their task).

Implementing a fully-connected feed forward network as final layer counters the problem of the information deprecation that is present in the [CLS] approach: Every token's BERT embedding gets concatenated with the token's SRL embeddding. The whole sequence is then flattened, i.e. all BERT+SRL vectors get concatenated into one large vector of size $\mathbb{R}^{n\times 768+20}$.

4.5.1.3 GRU Head

Since the SRLs are essentially a sequential "mark-up" of the sentences, the thought of encoding them with an architecture designed for sequential data is not too far. Inspired by the biological properties of the brain, the concept of recurrent neural networks has been around since the late 80ies, with [Hopfield, 1982] often being credited as having implemented the first recurrent neural network. To overcome the problems of the vanishing and explofing gradient problems, [Hochreiter and

Schmidhuber, 1997] proposed the LSTM architecture. [Cho et al., 2014] GRU

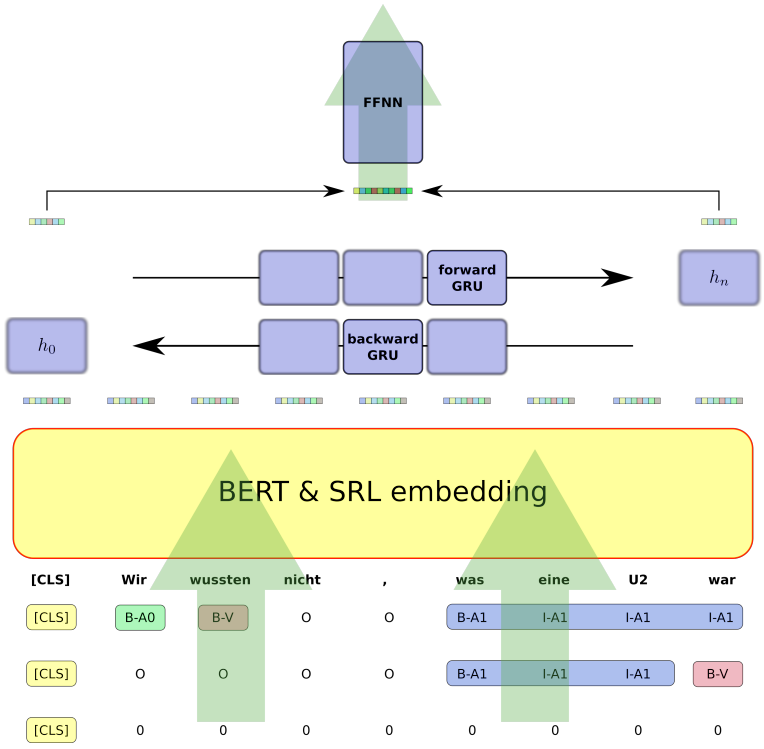


Figure 19: Head with a one-layer **F**eed **F**orward **N**eural **N**etwork on the concatenated last hidden states of the bi-directional GRU.

4.5.2 Question Answering

"[...] in the question answering task, we represent the input question and passage as a single packed sequence, with the question using the A embedding and the passage using the B embedding. We only introduce a start vector $S \in \mathbb{R}^H$ and an end vector $E \in \mathbb{R}^H$ during fine-tuning. The probability of word i being the start of the answer span is computed as a dot product between T_i and S followed by a softmax over all of the words in the paragraph: $P_i = \frac{e^{S \cdot T_i}}{\sum_j e^{S \cdot T_j}}$." [Devlin et al., 2018]

4.5.2.1 Span Prediction Head

The "standard" implementation of the Q&A BERT head consists of one simple FFNN which predicts on each input token the probability of it being the start of the answer span and the end of the answer span, respectively. After both probabilities are computed for all tokens, the ones with the highest probability get selected; no further logic is enforced, such as that the index of the start token should be smaller than the one of the end token, or that the answer span may not lie inside the question

(i. e. before the first [SEP] token), etc.

Since the standard implementation already implements a model that predicts on each token, the SRL-emgedding enriching is relatively straight forward: First, the input layer of the small head FFNN needs to be adapted to the BERT-token + SRL dimensionality, which results in the vector representation $\mathbb{R}^{768+20}$ for each SRL-enriched token. Secondly, when the setting of merging the BERT subtokens before adding the SRL embeddings is used, the start and end span indexes have to be recomputed.

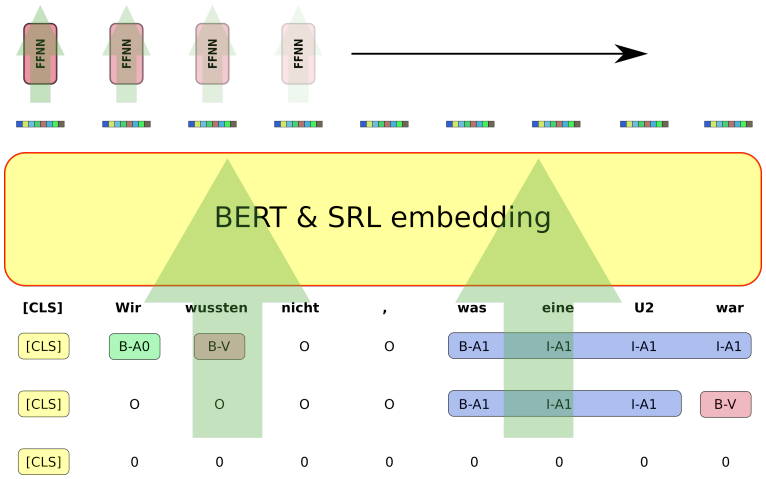


Figure 20: The GliBERT head for span prediction in Question Answering Tasks. After the Tokens and SRLs were consumed by BERT and the SRL embedding module, one FFNN predict on each token in the sequence, how likely it is first and the last token of the answer span. After these predictions have been made for all tokens, the token with the highest value for each position gets selected.

5 Results

In this chapter I report the outcomings of my experiments on the GerGLUE dataset. Because there were several different implementations of various parameters tested against each other (SRLs duplicated vs. SRLs zeroed, BERT tokens merged vs. SRLs splitted, etc.), I provide several aggregated "views" on the plain numerical results, where one or more of these implementations is paid attention to. Further, I document the ourcomings of a human assessment of the quality of the DAMESRL produced SRLs to be able to make propositions about the usefulness of those and their effect on the results. Similar to this, the general data quality of random samples of two datasets is manually reviewed, for the same reasons. Eventually, I carry out a small ablation study, controlling the effect of ghosting out different SRL structures to show that the actual, semanticity providing power of SRLs lies in the combination of these structures.

Since the different dataset types were tested with different heads, I group the them the following way: Results for classification datasets (deISEAR, SCARE, PAWS-X, XNLI) are reported together, and the results for question answering datasets (MLQA, XQuAD), are grouped together. Since the latter group consists of only two datasets addressing the same task, and bearing in mind that there were only experiments for one head in this group, the Span Prediction Head, the discussion analyzing these results will not be as interesting and substantial as for the first group. For space reaons, and for the sake of straightforwardness, only test set results are reported in the tables. However, model selection was always based on development set results (see figure fig:acc-loss for visualizations of development and test resulst compared).

5.0.1 Controlling for Statistical Significance

Before reporting the actual results, I briefly elaborate on my strategy for assessing statistical significance, since this is a crucial aspect of empirical analysis in general, and in attributing superiority of one model over the other on the basis of measured perfmoance on a dataset.

"if we rely on empirical evaluation to validate our hypotheses and reveal the correct language processing mechanisms, we better be sure that our results are not coincidental." [Dror et al., 2018]

$\delta(X) = M(A, X) - M(B, X)$

$H_0 : \delta(X) \leq 0$

$H_1 : \delta(X) > 0$

"It is important to have a method at hand that gives us assurances that the observed increase in the test score on a test set reflects true improvement in system quality." [Koehn, 2004]

Koehn [2004] focus strongly on significance testing in the context of evaluating on a sub-sample of the test set — due to expensiveness of testing on the whole set — and making statements about the reliability of this subset sample:

"Given a test result of m BLEU, we want to compute with a confidence q (or p-level P = 1 - q) that the rue BLEU score lies in an interval [m - d, m + d]." [Koehn, 2004]

Since the systems under review here predict on the exact same test set, the assumed independence of the predictions of the two models holds no longer. Morgan [2005] propose the following algorithm for testing difference significance:

"When the results are better with the new technique, a question arises as to whether these result differences are due to the new technique actually being better or just due to chance. Unfortunately, one usually cannot directly answer the question "what is the probability that the new technique is better given the results on the test data set"" [Yeh, 2000]

"But with statistics, one can answer the following proxy question: if the new technique was actually no different than the old technique (the null hypothesis), what is the probability that the results on the test set would be at least this skewed in the new technique's favor?" [Yeh, 2000]

Many evaluation metrics "have a tendency to underestimate the significance of the results", due to their inherent assumption that the compared systems "produce independent results" when in reality, they tend to produce "positively correlated results". [Yeh, 2000]

Algorithm 2 Approximate Randomization Algorithm

$p(M, x)$ = prediction of model M on example x
A, B = Two different models
$O = \{x_1, \ldots, x_n\}$ = test set
$O_A = \{p(A, x_1), \ldots, p(A, x_n)\}$
$O_B = \{p(B, x_1), \ldots, p(B, x_n)\}$
O_{gold} = gold labels for $\{x_1, \ldots, x_n\}$
$e(\hat{Y}, Y)$ = evaluation function for gold labels $\hat{Y}$ and predictions Y
$t_{original} = \mid e(O_{gold}, O_A) - e(O_{gold}, O_B) \mid$
$rand()$ = returns 0 or 1, randomly
$swap(x, y)$ = exchanges elements $x \in A, y \in B$ such that $y \in A, x \in B$
$r \leftarrow 0$
$R \leftarrow 0$
$threshold \leftarrow 1,000$
$p \leftarrow 0.05$
while $R < threshold$ **do**
  for all $(a_i, b_i) \in O_A \times O_B \mid i \in I$ **do**
    if $rand() = 0$ **then**
      $swap(a_i, b_i)$
    end if
  end for
  $t_{permute} = \mid e(O_{gold}, O'_A) - e(O_{gold}, O'_B) \mid$
  if $t_{permute} \geq t_{original}$ **then**
    $r \mathrel{+}= 1$
  end if
  $R \mathrel{+}= 1$
end while
if $\frac{r+1}{R+1} < p$ **then**
  system A truly better than system B
end if

5.0.1.1 Example Case for XNLI

Let's consider the case for the non-merged subtokens setting in the resampled XNLI dataset. The test set contains 1,125 sentence pairs for which textual entailment must be predicted. From these 1,125 sentence pairs, 398 bear the gold label *contradiction*, 357 are labeled *entailment*, and 370 are *neutral*; so, the class distribution of the set is fairly balanced.

I trained and optimized five systems for two architectures on the training and development set of XNLI: One architecture is the plain "vanilla" GRU classifier described in section XXX, the other is the same GRU architecture enriched with embedded SRLs (implementing the duplication approach, described in section XXX). The "vanilla" system ensemble achieved an accuracy of 66,58% on the XNLI test set, while the SRL enriched ensemble scored a 68,27% — in other words, the SRL enchriched ensemble performed 1,69% better than the "vanilla" ensemble.

To check if this difference truly measures the supremacy of the latter model over the first, I apply the above described algorithm 2 for testing significance by permuting the actual ensemble predictions. Note that both ensemble models were equally right or wrong in 1,018 cases out of 1,125. From this follows, in consequence, that in 90,49% of the cases the flipping of predictions between the ensemble models will have no effect.

Result p = 4.80%

In contrary, if we compare this results to the zero implementation of SRLs, we observe something different: The accuracy of this ensemble was slightly lower than the duplicate architecture; namely 67,73% or, speaking in differences, 1.15% better than the vanilly ensemble. The number of equally right or wrong examples was also slightly lower — 1,010 examples were equally wrong or correct predicted by the systems.

Result zeros p = 14.09%

In summary it is safe to say that although there is a positive effect of injecting SRL information during training over all datasets and architectures, this effect is arguably quite small and unsteady. this is especially in yontrast to Zhang et al. [2019b], who report more stable and higher effects In the next sections I will try to give an answer as to what are the reasons for these, honestly spoken, moderate results. Concretely, I will argue that this is mainly due to noise, present in differing intensities and at various levels in the data I acquired, that the model has to cope with:

Afterwards, statistical signifance is indicated by appending one, two, or three asterisks to a result that was compared with one or several others: * stands for $p < 10\%$, ** for $p < 5\%$, and *** indicates very high significance of $p < 1\%$.

5.1 Classification Dataset Results

- conjecture: SRLs are rather adding noise in sequences that are too long. Extreme exmples are the Q&A datasets.
- GRU architecture is probably strongest: most best models (even though mostly -SRL) and second best models, no worst performance
- 6 sgnificantly better +SRL vs. 3 significantly worse +SRL. There seems to be a slight trend that when merging subtokens, duplicating SRLs when too less predicates is better.
- all 3 worsening are for subtokenized architectures.

To obtain as stable results as possible, I decided to train five models for each architecture and configuration, all initialized with different random seeds. Additionally, I ensembled the five models, achieving a performance gain of several percentage points (see example of PAWS-X, table 8).[1] In table 7 below, the test set ensemble results for each architecture on the classification datasets are reported. For each row, i.e. for one hyperparameter set-up for one dataset, The best, second best and worst performace over all three heads and SRL configurations are marekd. In the last row, Best and second best results per head and +/-SRL for each subtokenization settings are accumulated. As the Scores indicate, taken the results without controlling for significance,[2] the +SRL configurations achieved 5 times the best, and 10 times the second best results — compared to 4 best and 5 second best results for -SRL. This is a first indicator that adding SRL information seems to be able to support plain BERT embeddings in classification tasks.

The accumulations in roman numerals also indicate a further, observable trend: The GRU head accumulated 5 best and 6 second best results, independent of with or

[1] For ensembling, a straightforward majority vote of the five models is implementd.

[2] Here I don't control for statistical significance because it is not clear, against what model(s) should a peak performance be controlled for: Take the 77.45% accuracy of the deISEAR α, achieved by the +SRL, BERT tokens subtokenized, FFNN head: Against what should this results be compared? To all other -SRL results? Only to the corresponding -SRL results for the same head?

without SRLs, appears to be better suited for classification tasks than the [CLS] (2 best, 6 second bset) and FFNN (2 best, 3 second bset) heads. Supporting this, the GRU head never was responsible for the worst performance, which were all achieved by the FFNN and [CLS] heads.

In the following table 8, the ensembling of each of the results in the main table is exemplary disaggregated. Each head-SRL configuration is randomly initialized 5 times, resulting in 5 models achieving a certain performance on the dataset. By ensembling them, a steady gain of, in this case, 1.26% compared to the average of the 5 models is achieved. Similar ensembling improvements where obseverd in all other datasets.

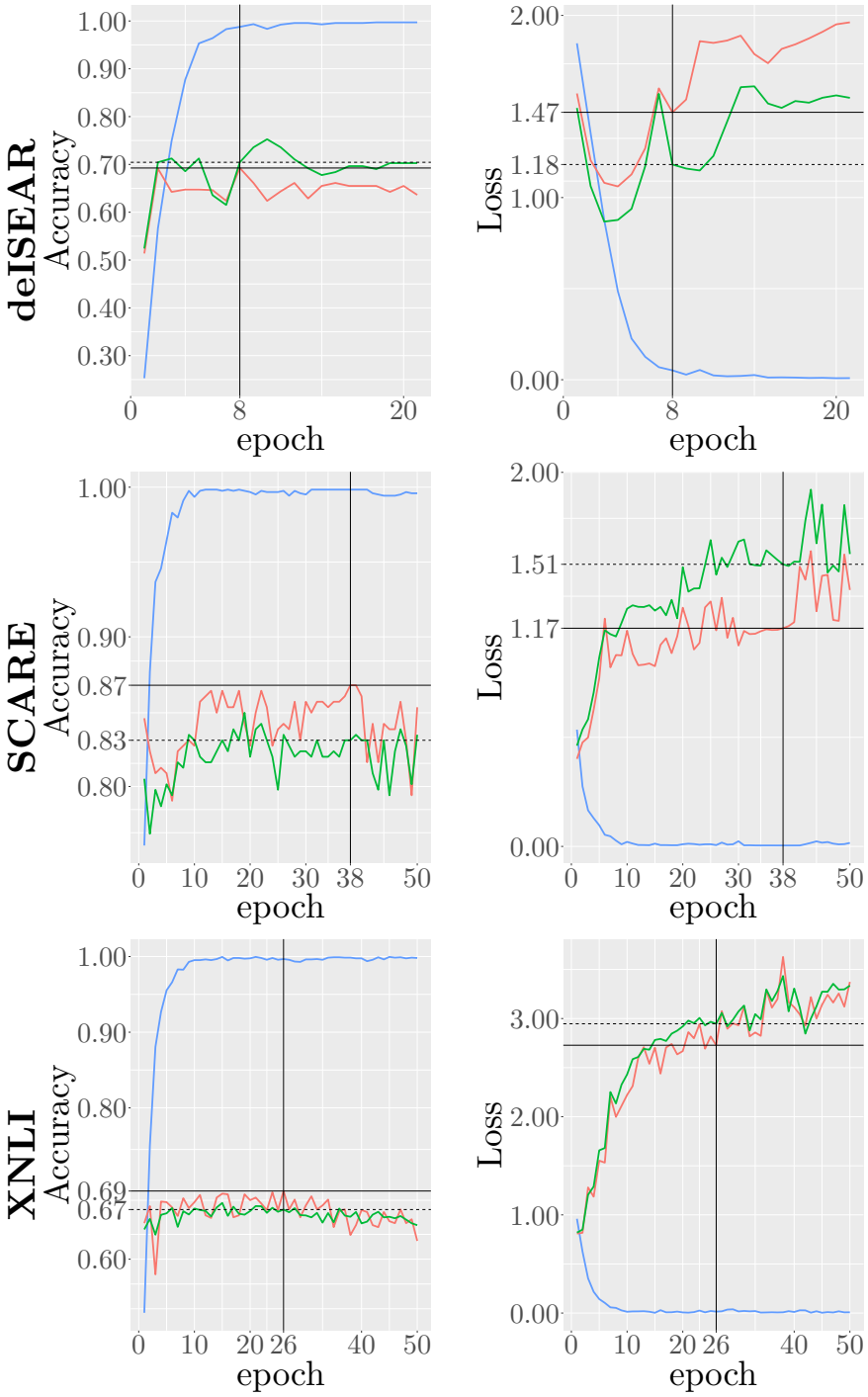


Figure 21: Accuracy and loss plots for three experiments (training, development, and test set): In the deISEAR plot, we see that early stopping was triggered after the loss on the development set increased for four contiguous epochs — indicating an overfitting of the model on the training set. Both SCARE and XNLI experiments ran until the maximum number of epochs were reached without triggering early stopping. However, the SCARE accuracy and loss progressions show much more fluctuations and erratic behavior than the XNLI ones, indicating the noisiness of the data and the randomness of the function that needs to be learned by the model.
The vertical line marks the epoch with the highest development accuracy, the dashed horizontal marks the corresponding test set value, and the drawn through the development value, respectively.

Classification Datasets

		[CLS] Head						**FFNN Head**						**GRU Head**					
		subtokenized			subtokens merged			subtokenized			subtokens merged			subtokenized			subtokens merged		
		−SRL	+SRL		−SRL	+SRL		−SRL	+SRL		−SRL	+SRL		−SRL	+SRL		−SRL	+SRL	
			zeros	dupl.		zeros	dupl.		zeros	dupl.		zeros	dupl.		zeros	dupl.		zeros	dupl.
deISEAR	α	71.52	72.19	71.52	72.19	*67.55*	72.19	70.86	**77.48**	72.85	74.17	72.85	74.17	70.20	74.83	74.17	73.51	70.20	71.52
	β	71.52	74.83	74.83	70.86	70.86	72.85	74.83	*68.21*	70.20	74.83	73.51	70.86	73.51	74.83	72.19	**76.82**	70.20	74.83
SCARE	α	85.61	82.58	83.71	83.33	83.71	85.61	83.33	83.71	84.09	84.09	*81.44*	84.47	83.71	83.33	84.09	**85.98**	84.09	83.71
	β	86.36	84.85	84.47	85.23	85.23	85.23	**86.74**	85.98	85.23	84.47	*83.33*	84.09	**86.74**	85.98	83.71	86.36	84.09	85.23
PAWS-X	α	80.63	81.60	81.49	*79.92*	80.63	82.51	81.19	80.78	80.07	80.43	80.02	80.68	82.26	82.77	82.77	82.82	82.87	**83.53**
	β	87.49	88.05	**88.21**	87.75	87.24	88.00	86.83	87.39	87.09	87.75	*86.58*	86.68	87.60	87.60	87.90	88.00	88.00	88.05
XNLI	α	67.34	**67.52**	66.64	66.94	66.94	*66.26*	67.20	67.42	67.34	66.38	67.08	66.92	66.68	66.60	67.14	66.42	66.54	66.26
	β	68.09	68.18	66.84	67.82	67.82	68.36	66.31	65.60	66.40	*64.98*	65.51	65.07	66.84	67.82	67.02	67.64	66.31	**68.53**
Scores		II	**II** II			II		**I** I	**I** I		I			**I**	II		**II** I	**II** III	

+SRL **5** 10

−SRL **4** 5

Table 7: Test set accuracy ensemble results (per 5 models) on single sentence and sentence pair tasks. **Bold** font marks the best result per line, underline the second best, and *italics* the poorest. In the *Scores* row, the afore mentioned positive extremes are accumulated for −SRL and +SRL; note that if both +SRL configurations of an architecture achieved an extreme, it is only counted once.
The line marked with light gray — PAWS-X β — is "expanded" in table 8 to illustrate that each result in this table is actually a majority voting out of an ensembeling of fice models.

PAWS-X β

	[CLS] Head						**FFNN Head**						**GRU Head**					
	subtokenized			subtokens merged			subtokenized			subtokens merged			subtokenized			subtokens merged		
	−SRL	+SRL		−SRL	+SRL		−SRL	+SRL		−SRL	+SRL		−SRL	+SRL		−SRL	+SRL	
		zeros	dupl.		zeros	dupl.		zeros	dupl.		zeros	dupl.		zeros	dupl.		zeros	dupl.
Model 1	85.41	85.36	86.02	86.43	86.53	87.04	85.77	85.61	85.77	86.22	84.70	86.53	87.54	87.49	86.43	85.51	86.93	86.53
Model 2	86.07	86.83	86.99	85.87	86.32	86.68	86.17	85.82	87.39	85.92	85.36	85.26	87.04	86.99	85.87	87.19	86.07	87.44
Model 3	86.07	87.49	86.99	87.14	85.26	86.73	86.22	84.90	85.36	85.41	85.31	85.71	86.12	85.66	86.83	86.48	87.09	86.93
Model 4	87.39	86.32	86.58	86.38	84.04	87.65	86.53	86.73	85.61	85.82	85.61	86.38	84.99	86.02	87.70	86.99	86.68	86.88
Model 5	86.63	86.43	86.58	86.99	85.26	85.56	86.18	87.09	84.75	87.04	85.77	85.82	86.12	85.82	86.73	87.34	86.73	86.58
Average	86.31	86.49	86.63	86.56	85.48	86.81	86.22	86.03	*85.78*	86.08	85.35	85.94	86.35	86.40	86.71	86.70	86.70	**86.87**
Ensemble	87.49	88.05	**88.21**	87.75	87.24	88.00	86.83	87.39	87.09	87.75	*86.58*	86.68	87.60	87.60	87.90	88.00	88.00	88.05
Gain	1.18	1.56	1.58	1.19	1.76	1.19	.65	1.36	1.31	1.67	1.23	.74	1.25	1.20	1.19	1.30	1.30	1.18
Average	1.26 (σ 0.28)																	

Table 8: The "expanded" PAWS-X β results. The light gray line corresponds to the one in table 7. As can be seen, the fluctuations between single models is not too big, which is an indicator that the architecture is fairly stable. Ensembeling reliably leads to a 1.26 percentage points gain on average, with a standard deviation of 0.26%.

To see whether there can be seen a tendency as to which SRL-BERT aligning strategy turns out to be more effective, the results of the main table 7 are aggregated in table 9: For each head, the +SRL results are compared, and the differences are reoprted in the table. For example, we look at the [CLS] head on deISEAR α and ask us, which strategy — merging the BERT subtokens back ot "normal" tokens, or splitting the SRLs up to align with the subtokenized BERT tokens — worked better. For this, the difference between both for +SRL zeros and +SRL duplicate is calculated and controlled for statistical signifcance between both ensembles. The subtokenizing strategy (yellow) outperformed the merging strategy (purple) 28 times, 15 times of it statistically significant, while merging was better 20 times, only 15% of it significantly. Therfore, at least for the GerGLU classification datasets, splitting the SRLs up to align with the BERT subtokens seems more effective than merging the BERT subtokens. This also makes sense intuitively — using the merging strategy of averaging the BERT embeddings of subtokens leads clearly to an information loss or distortion that cannot be fully balanced by the added SRL information.

		[CLS] Head		**FFNN Head**		**GRU Head**	
		zeros	dupl.	zeros	dupl.	zeros	dupl.
deISEAR	α	4.64**	.67	4.63*	1.32	4.63*	2.65
	β	3.97*	1.98	5.30*	.66	4.63*	2.64
SCARE	α	1.13	1.90	2.27*	.38	.76	.38
	β	.38	.76	2.65**	1.14	1.89	1.52
PAWS-X	α	.97*	1.02**	.76	.61	.10	.76
	β	.81**	.21	.81*	.41	.40	.15
XNLI	α	.58**	.38	.34	.42	.06	.88**
	β	.36	1.52*	.09	1.33*	1.51*	1.51*

Table 9: Performance of architectures when BERT subtokenized vs. merged. Both SRL implementations were compared pairwise. Example case, upper left 4.64**: Comparison of deISEAR α, +SRL zero implementation, [CLS] Head, subtokenized ensemble (72.19% accuracy) and the merged ensemble (67.55% accuracy) — the subtokenized ensemble performed 4.64% better, apparently with quite high significance ($p < 5\%$).

Another accumulating view on the main table 7 is constructed by not just looking for the best results ov a whole row, i.e. over all heads, merging strategies, and SRL implementations, but instead record the superiority or inferiority of +SRL compared to −SRL for each pairing: Take for example the deISEAR α subtokenized [CLS] head setting: Without SRLs, the ensemble performance was 71.52%; now the better of the two +SRL implementations of this setting is taken into account, in this case zeroing apparently was better. The difference, .67% is controlled for significance and reported in the table. The coloring indicates which SRL implementation performed better and was taken into account; sometimes both implementations permormed

		# of epochs	split set up	batch size	maximum length	SRL implementation
deISEAR	α	100	normal	16	40	normal
	β	100	normal	16	200	normal
SCARE	α	50	normal	16	50	normal
	β	50	normal	16	100	normal
PAWS-X	α	20	normal	16	16	normal
	β	20	normal	16	16	normal
XNLI	α	50	normal	16	100	normal
	β	50	re-split	16	100	normal
MLQA	α	40	200	50	100	normal
	β	40	200	50	100	normal
XQuAD	α	40	200	50	100	100
	β	40	200	50	100	100

Table 10: The different hyperparameter configurations for each dataset.

learning rate	2e-05
SRL embedding dimensions	20
SRL GRU hidden size	32
SRL number of layers	2
SRL bias	True
SRL bidirectional	True
SRL dropout	0.1

Table 11: General hyperparameter configurations.

equally, which is also reflected in coloring. Negative numbers indicate, of course, that the head produced better results when SRL information was *not* added (and this was also controlled for significance). Looking at the such accumulated results (table **??**), no clear picture emerges at first glance; sometimes zeroing SRLs performed better than duplicating SRLs, sometimes not, sometimes +SRLs clearly outperforms −SRL, even highly statistically, sometimes it is the other way around adding SRLs seems to create a disadvantage for the head.

To add one more abstraction layer and see a more complete picture, this table is again summarized in figure **??**. Doing this, the image clkarifies, allowing to stipulate that adding SRLs on GerGLUE classification tasks leads to a measurable improvement. The total gains clearly outweigh the losses, and also more statistically signifacnt. Regarding the SRL implementation, ther is no clearadvantage of one method over the other, only a slight tendency of duplicating being more effective than zeroing.

		[CLS] Head		FFNN Head		GRU Head	
		subtok.	merged	subtok.	merged	subtok.	merged
deISEAR	α	.67	.00	6.62**	.00	4.63**	−1.99
	β	3.31	1.99	−4.63*	−1.32	1.32	−1.99
SCARE	α	−1.90*	2.28*	.76	.38	.38	−1.89
	β	−1.51**	.00	−.76	−.38	.76	−1.13
PAWS-X	α	.97*	2.59***	−.41	.25	.51	.71
	β	.72*	.25	.56	−1.07***	.30	.05
XNLI	α	.18	−.22	.22	.70*	.46	.12
	β	.09	.44	.09	.53	.98	.89

Table 12: Ensemble percentage points gains (positive numbers) / losses (negative numbers) for +SRL over −SRL for each configuration from table 7. The better of the +SRL configurations was taken into account: zeros, duplicate. Light blue denotes that both architectures performed equally (in which case both ensembles were controlled for significance). One asterisk signifies a p-value $< 10\%$, two stand for $p < 5\%$ and three for $p < 1\%$.

As described in chapter 3, the datasets comiled in GerGLUE are a quite heterogenous assemble: Some datasets like deISEAR and, especially, SCARE comprise rather informal, colloquial tests, while PAWS-X and XNLI consits of more technical, standardized text. Further, as depicted in figure **??**, over 30% of the sentences in SCARE don't have any predicate-argument structures — in other words there is no exploitable SRL information for the model.

		Predicted		
		Positive	Neutral	Negative
True	Positive	149	12	5
	Neutral	4	11	6
	Negative	3	8	66

Table 13: Confusion matrix for SCARE α merged, +SRL duplicated. (macro F1: 73.52).

5.2 Question Answering Dataset Results

register noise The textual styles vary greatly from iutilizing complex, hypotactic sentence structures (e.g. XQuAD), to highly informal, elliptic — even erratic — structures (e.g. SCARE).

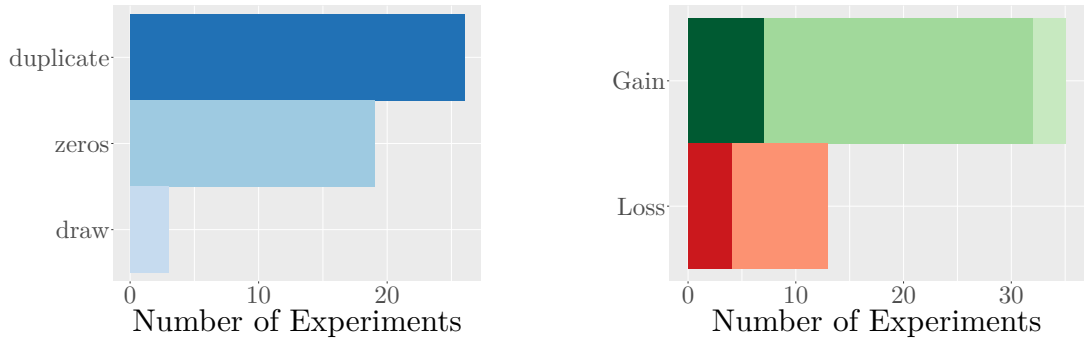


Figure 22: Accumulated scores from table 12. **Left**: Which SRL mode performed stronger out of a total of 48 settings. The bars indicate a slight outperforming of duplicating SRLs instead of adding zeros. **Right**: Counts for Gains and Losses in all 48 settings. Darker shades indicate significant results. The light green tip stands for settings where the gain equals 0.00.

label noise Many of my datasets were constructed either automatically (e.g. scrambling text automatically to create paraphrase pairs) or employing crowd-sourcing techniques. Either way, the process is prone to erros. There are, e.g., 84 sentence pairs in the trainig set of PAWS-X that are 100% identical, yet labelled as non-paraphrases.

translation noise Due to the mostly employed semi-automatic translation approach for creating the various datasets, errors have been introduced into the data ranging from typical translation errors (e.g. English "bishop" in the clerical context translated to the German chess figure counterpart "Läufer", not "Bischof") to eventually wrongly copied labels, since the overall meaning changed during the tranlsation process (e.g. a sentence pair is no more contradictive but neutral).

SRL noise The SRLs obtained from DAMESRL are, conservatively formulated, questionable in their quality (e.g. modifiers are completely missing).

In short — the old GIGO concept from informatics holds mutatis mutandis also in NLP.

5.3 Register Noise

5.4 Label Noise

As [Caswell et al., 2021] point out,

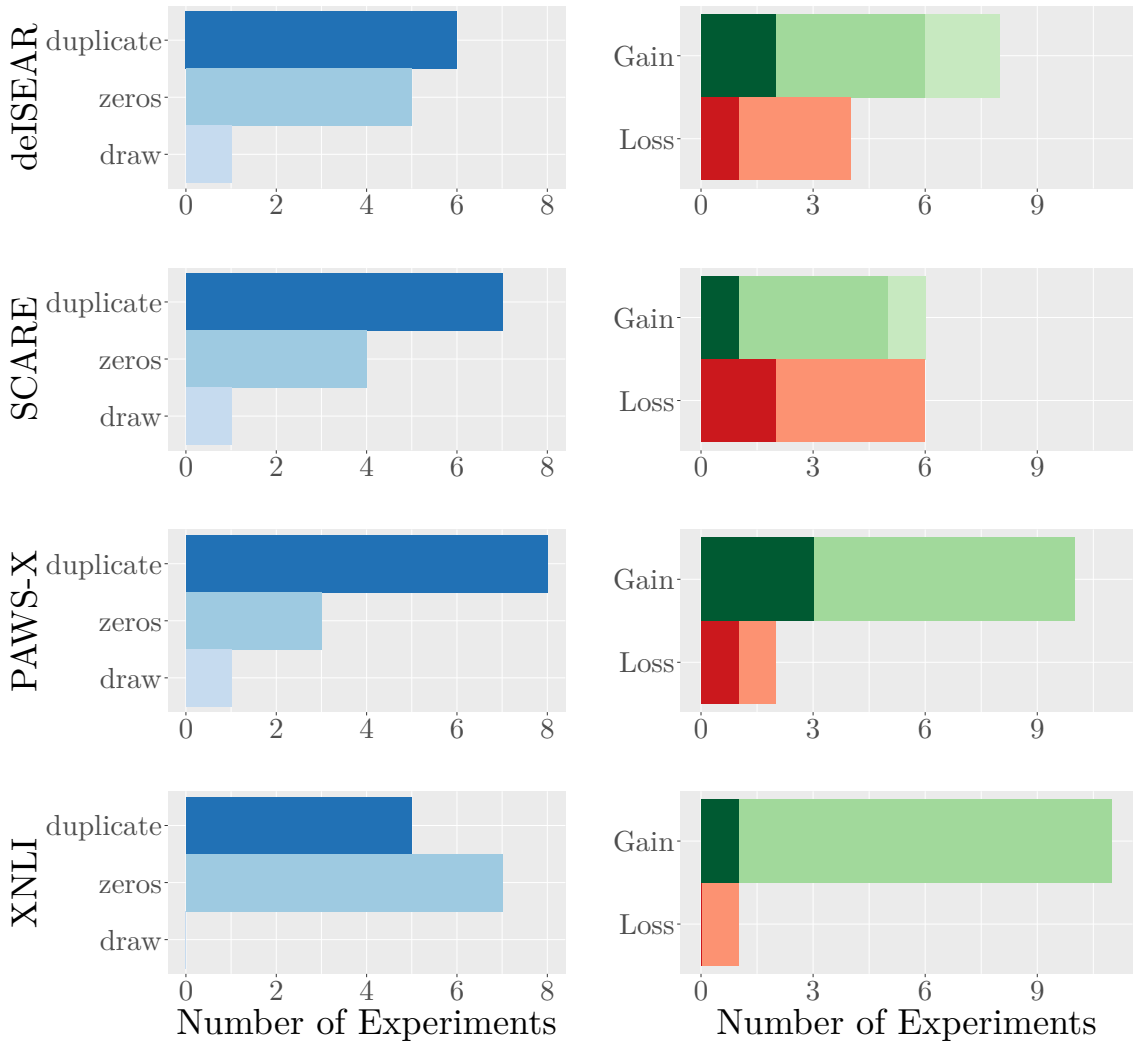


Figure 23: Accumulation of statistics for each classification datasetclassification. **Left**: Which SRL architecture performed better. **Right**: Comparison of accuray points gained/lost after adding SRLs.

PAWS-X sentence number 45061, labelled as non-paraphrases:

Riverton was a parliamentary electorate in the New Zealand region of Southland . Riverton was a parliamentary electorate in the New Zealand region of Southland .

Riverton war ein Parlamentswähler in der neuseeländischen Region Southland. Riverton war ein Parlamentswähler in der neuseeländischen Region Southland.

5.4.0.1 Re-annotation

PAWS-X

Q&A Datasets

		Span Prediction Head					
		subtokenized			subtokens merged		
		−SRL	+SRL		−SRL	+SRL	
			zeros	dupl.		zeros	dupl.
MLQA	α	**30.69**	29.68	29.68	21.92	21.92	*21.81*
	β	**44.75**	44.55	43.41	*41.66*	41.86	41.79
XQuAD	α	**42.01**	41.42	41.12	37.87	36.98	*35.50*
	β	46.57	45.43	**46.86**	37.43	*37.14*	39.14
Scores		**III** I		**I** III			
+SRL	**1** 3						
−SRL	**3** 1						

Table 14: Test set accuracy ensemble results (per 5 models) on question answering tasks. **Bold** font marks the best result per line, underline the second best, and *italics* the poorest.

2 human annotators re-label 20 examples of PAWS-X where gold != predicted. Fleiss' κ between 2 annotators: 0.68 Fleiss' κ between 2 annotators and gold: 0.3541

6 examples where both annotators agree with predictions, disagree with gold:

(5.1) Der NVIDIA TITAN V wurde von Nvidia am 7. Dezember 2017 offiziell angekündigt.
Am 07. Dezember 2017, verkündete NVIDIA offiziell Nvidia TITAN V.

humans & model: False, Gold: True

(5.2) Die Schäfte sind sehr kurz oder oft nicht vorhanden.
Es sind entweder wenig Landschaften vorhanden oder sie fehlen in den meisten Fällen.

humans & model: False, Gold: True

(5.3) 1963 trat Roy der Kommunistischen Partei Indiens bei und leitete Gewerkschaftsbewegungen in Bansdroni in Kalkutta.
Roy trat 1963 der Kommunistischen Partei Indiens bei und leitete Gewerkschaftsbewegungen im Kolkata-Gebiet von Bansdroni.

humans & model: True, Gold:False

(5.4) Der Kanal ist einer der ältesten schiffbaren Kanäle Europas und sogar

		Span Prediction Head			
		Gains/Losses		subtokenized/merged	
		subtok.	merged	zeros	dupl.
MLQA	α	−1.01***	.00	7.76***	7.86***
	β	−.20	.20	2.69	1.62
XQuAD	α	−.59	-.89	4.44**	5.62***
	β	.29	1.71	8.29***	7.72***

Table 15: Left part: Ensemble percentage points gains (positive numbers) / losses (negative numbers) for +SRL over −SRL for the Span Prediction Head from table 14. The better of the +SRL configurations was taken into account: zeros, duplicate. Light blue denotes that both architectures performed equally (in which case both ensembles were controlled for significance). One asterisk signifies a p-value $< 10\%$, two stand for $p < 5\%$ and three for $p < 1\%$. **Right part**: Performance of architectures when BERT subtokenized vs. merged. Both SRL implementations were compared pairwise..

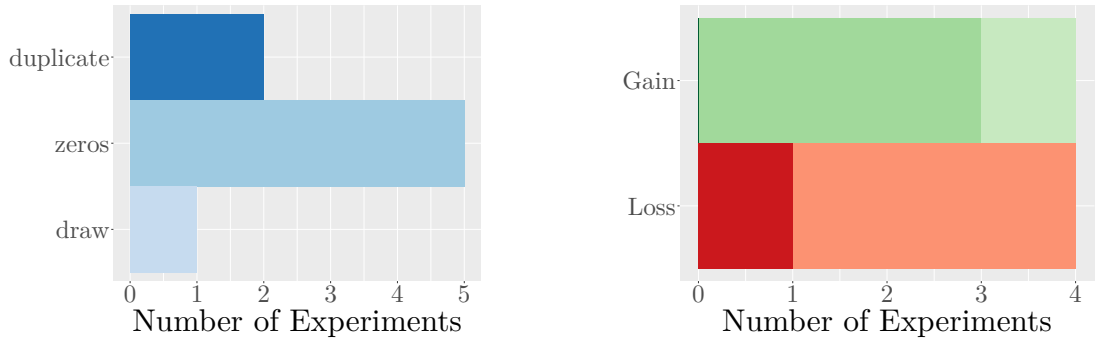


Figure 24: Accumulated scores from table **??**. **Left**: Which SRL mode performed stronger out of a total of 8 settings. The bars indicate a slight outperforming of duplicating SRLs instead of adding zeros. **Right**: Counts for Gains and Losses in all 8 settings. Darker shades indicate significant results. The light green tip stands for settings where the gain equals 0.00.

Belgiens.
Der Kanal ist einer der ältesten befahrbaren Kanäle in Belgien und Europa.

humans & model: True, Gold:False

(5.5) Propilidium pelseneeri ist eine Art der Meeresschnecken, eine wahre Napfschnecke und Gastropoden-Mollusk in der Familie der Lepetidae.
Propilidium pelseneeri ist eine Art der Meeresschnecken, eine wahre Napfschnecke und Meeres-Gastropoden-Mollusk der Familie der Lepetidae.

humans & model: True, Gold:False

(5.6) Die Chicago Bears sanken auf die Giants 27:21, und verloren 0:6 zum ersten

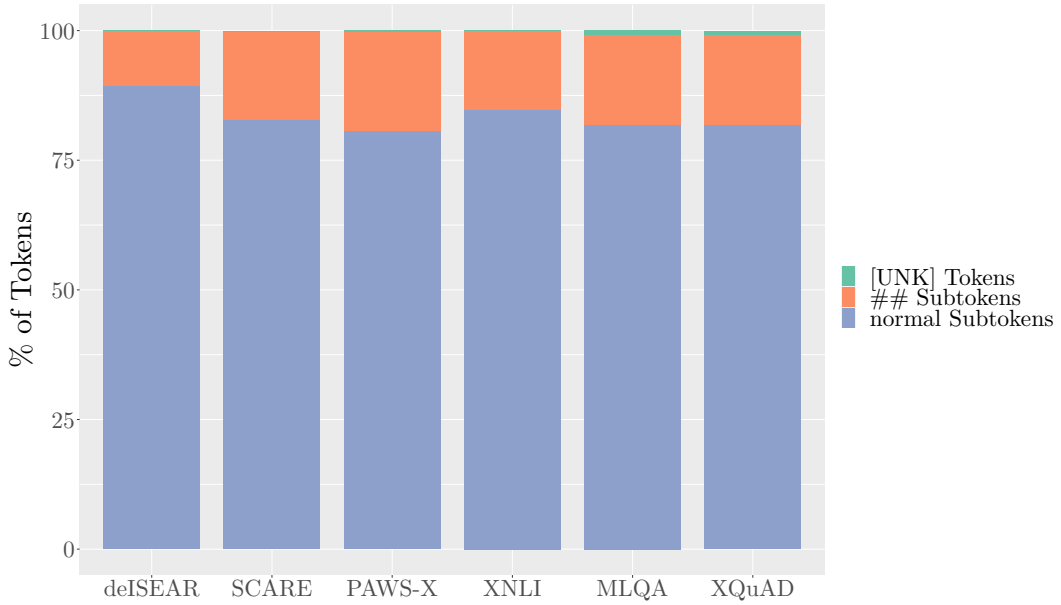


Figure 25: Percentages of token-types in all datasets. ## Subtokens represent the amount of tokens that get re-merged in the merged settings (e.g. "Master" "##arbeit" → "Masterarbeit". The amount of tokens that lie outside of the German BERT vocabulary is in all datasets extremely small (for deISEAR and XNLI there are no [UNK]s at all); the largest shares of such tokens are present in MLQA and XQuAD with .79% and .73%, respectively.

Mal seit 1976.
Die Chicago Bears verloren 21:27 gegen die Bears und standen erstmals seit 1976 bei 0:6.

humans & model: False, Gold:True

XNLI

2 human annotators re-label 20 examples of XNLI where gold != predicted. Fleiss' κ between 2 annotators: 0.36 Fleiss' κ between 2 annotators and gold: 0.4787

1 example where 2 humans == model and 2 humans != Gold

Bato ist ein Jahrhunderte altes Wort, das man als Kerl oder Kumpel übersetzen kann. Bato (oder Vato) ist ein spanisches Wort, das Typ oder Typ bedeutet.

humans & model: neutral, Gold: Entailment

1 example where 2 humans != model and 2 humans != Gold

Oh, ich sehe oh der Staat braucht es nicht gut, das ist eher das, das ist eher

ungewöhnlich, nicht wahr? Das macht Sinn, dass der Staat es benötigt.

humans: neutral, model: entailment, Gold: contradiction

5.5 Translation Noise

XNLI labelled as entailment

and that's a lot of it is due to the fact that the mothers are on drugs The mothers take drugs.

Und vieles davon liegt daran, dass die Mütter Medikamente nehmen. Die Mütter nehmen Drogen.

PAWS-X; different repair-strategies → different labels (gold: false)

Sawyers autorisierte Biografie wurde 2014 von Huston Smith veröffentlicht. Im Jahr 2014 wurde Huston Smith eine autorisierte Biographie von Sawyer veröffentlicht.

Im Jahr 2014 wurde «Huston Smith», eine autorisierte Biographie von Sawyer, veröffentlicht.

Im Jahr 2014 wurde von|für|durch|trotz|wegen Huston Smith eine autorisierte Biographie von Sawyer veröffentlicht.

Im Jahr 2014 wurde Huston Smith eine autorisierte Biographie von Sawyer veröffentlicht.

5.6 SRL Noise

A major question arising in the context of using automatically assigned Semantic Roles in downstream tasks, is how good these Semantic Roles are. Since there is no gold standard available for Semantic Role Labels for the datasets I use in my experiments, there is no straight-forward way to evaluate their quality automatically. In contrast to other tagging tasks like POS prediction or NER, Semantic Roles are not as black and white: While it is relatively easy to decide if a predicted POS tag is correct or incorrect, it is more a scale concerning SRLs.

Fleiss' $\kappa = 0.2048$ — this slightly above the threshold of «fair agreement», as defined by [Landis and Koch, 1977] (0.20).

The κ for helpful vs. other is even worse: 0.1944

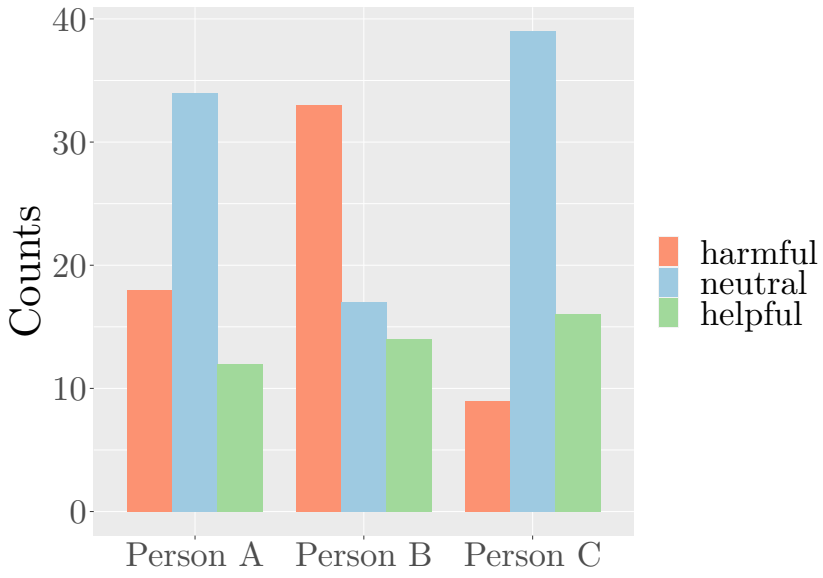


Figure 26: Independent evaluation of SRL quality by three people. Regardless of the label attributed to each example, it is obvious, that the total amount of sentences for which the annotators evaluated the corresponding semanti roles as *helpful*, is relatively stable.

for individual datasets:

deISEAR: 0.0814

SCARE: 0.2401

PAWS-X: 0.1245

XNLI: 0.2475

MLQA: -0.3636

XQuAD: -0.5

Do et al. [2018]

5.7 Ablation study

To be able to make substantial claims about the positive influence about a new algorithm over an established one, it is common ground to conduct an ablation study. In such a study, one tries to determine which aspects of the proposed architecture contribute how much to the overall performance gain (or loss, respectively).

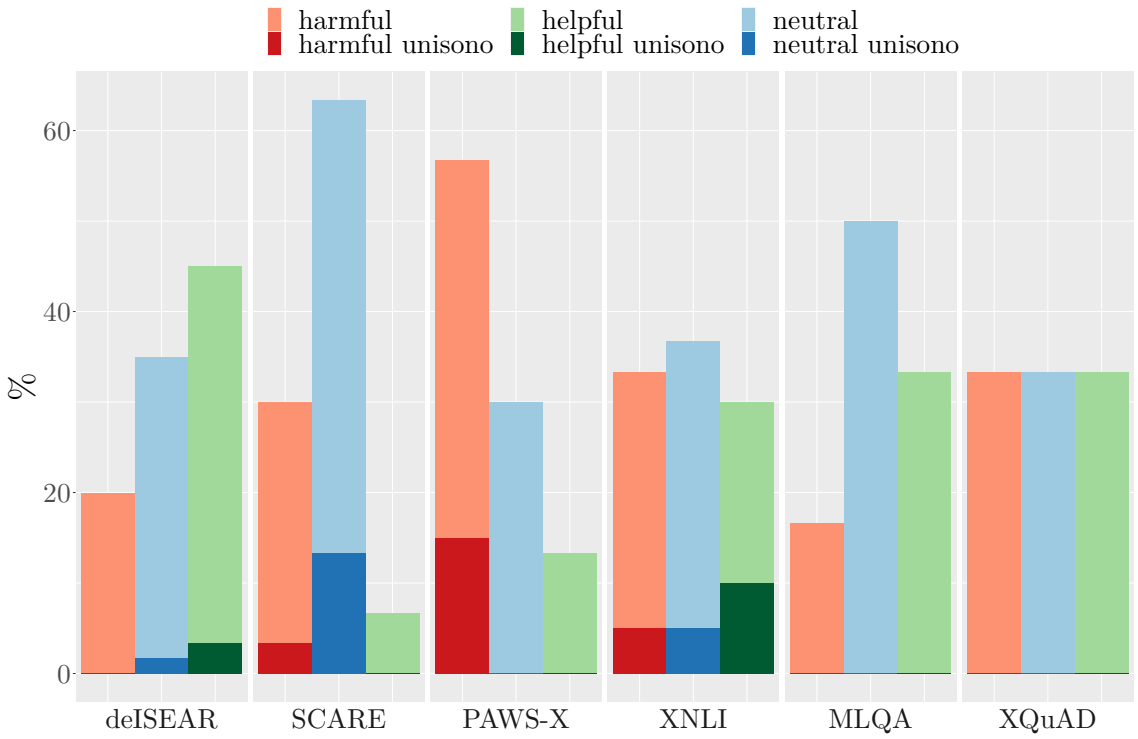


Figure 27: Estimated quality of SRLs per dataset.

In my case, i.e. the attempt to improve the performance of BERT regarding NLU tasks, the following question would need some ablation experiments to be answered: What part of the SRLs is most responsible for the performance boost? To be able to formulate this in a matter which can be experimentally tested, I identify two easily separatable and testable aspects of SRLs: Firstly, the information what parts of a sentence are the predicates. The intuition behind this is that maybe the head relies mostly on the information as to which tokens carry information about the events that happen in a given sentence. To test this, I simply drpo the information about all SRLs, except the information that a token is a predicate. In the second case, the hypothesis is reversed: Maybe the head is able to get the most useful hints about the information which inidcates what role certain token groups play in a given sentence. To test for this all information about predicates is dropped and only information about arguments is preserved.

Ich	B-A0	0
weiß	B-V	0
nicht	0	0
ob	B-A1	0
er	I-A1	B-A0
danach	I-A1	0
in	I-A1	B-A1
Augusta	I-A1	I-A1
geblieben	I-A1	B-V
ist	I-A1	0
.	0	0
==============================		
Er	B-A0	
wohnte	B-V	
weiterhin	0	
in	B-A1	
Augusta	I-A1	
.	0	

SRL 5.1: Normal SRLs.

Ich	0	0
weiß	B-V	0
nicht	0	0
ob	0	0
er	0	0
danach	0	0
in	0	0
Augusta	0	0
geblieben	0	B-V
ist	0	0
.	0	0
==============================		
Er	0	
wohnte	B-V	
weiterhin	0	
in	0	
Augusta	0	
.	0	

Ich	B-A0	0
weiß	0	0
nicht	0	0
ob	B-A1	0
er	I-A1	B-A0
danach	I-A1	0
in	I-A1	B-A1
Augusta	I-A1	I-A1
geblieben	I-A1	0
ist	I-A1	0
.	0	0
==============================		
Er	B-A0	
wohnte	0	
weiterhin	0	
in	B-A1	
Augusta	I-A1	
.	0	

SRL 5.2: Left: Only predicate SRLs. **Right**: Only argument SRLs.

		−SRL	+SRL		
			only PREDs	only ARGs	normal
deISEAR α	FFNN Head subtok. zeros	*70.86*	72.19	75.50**	**77.48****
SCARE α	[CLS] Head merged duplicate	*83.33*	84.47	85.23	**85.61***
PAWS-X β	[CLS] Head merged duplicate	*79.92*	80.53	80.68	**82.51*****
XNLI β	GRU Head subtok. zeros	*66.84*	67.02	**68.00**	67.82

Table 16: Ablation on Effect of PREDs and ARGs isolated. note that PRED/ARG SRL not significant (SCARE, XNLI ARGs almost, ca. 11%)

6 Conclusion

In a first paragraph, I will quickly point out the core elements of my thesis. In a second part, I locate my findings in the current debate; and lastly, I will look into further steps that could be taken from this point on.

My experiment consists mainly of three components: (1) The compilation of a German NLU dataset, *GerGLUE*, incorporating different tasks and modalities, as well as including domain-specific language, ranging from colloquial to highly stylized texts. (2) A pipeline which brings all data sets in the same workable format, computes SRLs by implementing two freely available tools (ParZu, DAMESRL) and several instruments for training and analyzing models. (3) *GliBERT*, a BERT-derived architecture which combines vanilla BERT-embeddings with embedded SRLs during fine-tuning, with several heads on top.

After carrying out a multitude of experiments with different heads, SRL implementations, and merging techniques, there is a slight tendence visible: For classification tasks injecting SRLs during finetuning seems to boost vanilla BERT embeddings to some degree. However, it has to be kept in mind that there was noise detected on several levels of the process, so that hypothetically, with cleaner data and better SRLs, the effect could be even stronger. For the both question answering tasks, there was no positive effect of SRL information observable. Generally,

6.1 Outlook / Future Work

- better SRL system (bspw. mit MOD-NEG detection)
- "sincere" hyper-param search
- more thoroughful SRL quality estimation
- compile data sets targeted at weaknesses of BERT
- "Manually" add negation particle from parse tree information

Glossary

Of course there are plenty of glossaries out there! One (not too serious) example is the online MT glossary of Kevin Knight [1] in which MT itself is defined as

> techniques for allowing construction workers and architects from all over the world to communicate better with each other so they can get back to work on that really tall tower.

accuracy A basic score for evaluating automatic **annotation tools** such as **parsers** or **part-of-speech taggers**. It is equal to the number of **tokens** correctly tagged, divided by the total number of tokens. [...]. (See **precision and recall**.)

clitic A morpheme that has the syntactic characteristics of a word, but is phonologically and lexically bound to another word, for example *n't* in the word *hasn't*. Possessive forms can also be clitics, e.g. The dog*'s* dinner. When **part-of-speech tagging** is carried out on a corpus, clitics are often separated from the word they are joined to.

[1] Machine Translation Glossary (Kevin Knight): `http://www.isi.edu/natural-language/people/dvl.html`

References

N. Aepli. *Parsing Approaches for Swiss German.* PhD thesis, University of Zurich, 2018.

M. Artetxe, S. Ruder, and D. Yogatama. On the cross-lingual transferability of monolingual representations. *arXiv preprint arXiv:1910.11856*, 2019.

S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, and Z. Ives. Dbpedia: A nucleus for a web of open data. In *The semantic web*, pages 722–735. Springer, 2007.

E. M. Bender, T. Gebru, A. McMillan-Major, and S. Shmitchell. On the dangers of stochastic parrots: Can language models be too big? In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, pages 610–623, 2021.

Y. Bengio, R. Ducharme, P. Vincent, and C. Janvin. A neural probabilistic language model. *The journal of machine learning research*, 3:1137–1155, 2003.

R. C. Berwick, N. Chomsky, and M. Piattelli-Palmarini. Poverty of the stimulus stands: Why recent challenges fail. *Rich languages from poor inputs*, pages 19–42, 2013.

Y. Bisk, A. Holtzman, J. Thomason, J. Andreas, Y. Bengio, J. Chai, M. Lapata, A. Lazaridou, J. May, A. Nisnevich, et al. Experience grounds language. *arXiv preprint arXiv:2004.10151*, 2020.

C. Bonial, J. Hwang, J. Bonn, K. Conger, O. Babko-Malaya, and M. Palmer. English propbank annotation guidelines. *Center for Computational Language and Education Research Institute of Cognitive Science University of Colorado at Boulder*, 48, 2012.

C. Bonial, J. Bonn, K. Conger, J. D. Hwang, and M. Palmer. Propbank: Semantics of new predicate types. In *LREC*, pages 3013–3019. Citeseer, 2014.

S. R. Bowman, G. Angeli, C. Potts, and C. D. Manning. The snli corpus. 2015.

E. Brill and R. J. Mooney. An overview of empirical natural language processing. *AI magazine*, 18(4):13–13, 1997.

T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020.

S. Buchholz and E. Marsi. Conll-x shared task on multilingual dependency parsing. In *Proceedings of the tenth conference on computational natural language learning (CoNLL-X)*, pages 149–164, 2006.

I. Caswell, J. Kreutzer, L. Wang, A. Wahab, D. van Esch, N. Ulzii-Orshikh, A. Tapo, N. Subramani, A. Sokolov, C. Sikasote, M. Setyawan, S. Sarin, S. Samb, B. Sagot, C. Rivera, A. Rios, I. Papadimitriou, S. Osei, P. J. O. Suárez, I. Orife, K. Ogueji, R. A. Niyongabo, T. Q. Nguyen, M. Müller, A. Müller, S. H. Muhammad, N. Muhammad, A. Mnyakeni, J. Mirzakhalov, T. Matangira, C. Leong, N. Lawson, S. Kudugunta, Y. Jernite, M. Jenny, O. Firat, B. F. P. Dossou, S. Dlamini, N. de Silva, S. Çabuk Ballı, S. Biderman, A. Battisti, A. Baruwa, A. Bapna, P. Baljekar, I. A. Azime, A. Awokoya, D. Ataman, O. Ahia, O. Ahia, S. Agrawal, and M. Adeyemi. Quality at a glance: An audit of web-crawled multilingual datasets, 2021.

I. Chalkidis, M. Fergadiotis, P. Malakasiotis, N. Aletras, and I. Androutsopoulos. Legal-bert: The muppets straight out of law school, 2020.

D. Chen and W.-t. Yih. Open-domain question answering. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts*, pages 34–37, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-tutorials.8. URL `https://www.aclweb.org/anthology/2020.acl-tutorials.8`.

K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

N. Chomsky. On cognitive structures and their development: A reply to piaget. *Language and Learning: The debate between Jean Piaget and Noam Chomsky*, 1980.

J. Chung, C. Gulcehre, K. Cho, and Y. Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.

A. Conneau, G. Lample, R. Rinott, A. Williams, S. R. Bowman, H. Schwenk, and V. Stoyanov. Xnli: Evaluating cross-lingual sentence representations. *arXiv preprint arXiv:1809.05053*, 2018.

M.-C. De Marneffe, T. Dozat, N. Silveira, K. Haverinen, F. Ginter, J. Nivre, and C. D. Manning. Universal stanford dependencies: A cross-linguistic typology. In *LREC*, volume 14, pages 4585–4592, 2014.

W. de Vries, A. van Cranenburgh, A. Bisazza, T. Caselli, G. van Noord, and M. Nissim. Bertje: A dutch bert model. *arXiv preprint arXiv:1912.09582*, 2019.

J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Q. N. T. Do, A. Leeuwenberg, G. Heyman, and M. F. Moens. A flexible and easy-to-use semantic role labeling framework for different languages. In *Proceedings of the 27th International Conference on Computational Linguistics: System Demonstrations*, pages 161–165, 2018.

D. Dowty. Thematic proto-roles and argument selection. *language*, 67(3):547–619, 1991.

R. Dror, G. Baumer, S. Shlomov, and R. Reichart. The hitchhiker's guide to testing statistical significance in natural language processing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1383–1392, 2018.

A. Ettinger. What bert is not: Lessons from a new suite of psycholinguistic diagnostics for language models. *Transactions of the Association for Computational Linguistics*, 8:34–48, 2020.

C. Fillmore. The case for case. 1967.

L. Floridi and M. Chiriatti. Gpt-3: Its nature, scope, limits, and consequences. *Minds and Machines*, 30(4):681–694, 2020.

K. A. Foth. Eine umfassende constraint-dependenz-grammatik des deutschen. 2006.

G. Furnas et al. Using latent semantic analysis to improve information retrieval. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*, pages 281–285. ACM Press, 1988.

K. Gerdes and S. Kahane. Word order in german: A formal dependency grammar using a topological hierarchy. In *Proceedings of the 39th annual meeting of the Association for Computational Linguistics*, pages 220–227, 2001.

D. Gildea and D. Jurafsky. Automatic labeling of semantic roles. *Computational linguistics*, 28(3):245–288, 2002.

A. Go, R. Bhayani, and L. Huang. Twitter sentiment classification using distant supervision. *CS224N project report, Stanford*, 1(12):2009, 2009.

T. Groß and T. Osborne. The dependency status of function words: Auxiliaries. In *Proceedings of the Third International Conference on Dependency Linguistics (Depling 2015)*, pages 111–120, 2015.

J. Hajič, M. Ciaramita, R. Johansson, D. Kawahara, M. A. Martí, L. Màrquez, A. Meyers, J. Nivre, S. Padó, J. Štepánek, et al. The conll-2009 shared task: Syntactic and semantic dependencies in multiple languages. 2009.

B. Hamp and H. Feldweg. Germanet-a lexical-semantic net for german. In *Automatic information extraction and building of lexical semantic resources for NLP applications*, 1997.

Z. S. Harris. Distributional structure. *Word*, 10(2-3):146–162, 1954.

P. He, X. Liu, J. Gao, and W. Chen. Deberta: Decoding-enhanced bert with disentangled attention. *arXiv preprint arXiv:2006.03654*, 2020.

S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

J. J. Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the national academy of sciences*, 79(8): 2554–2558, 1982.

R. S. Jackendoff. Semantic interpretation in generative grammar. 1972.

Y. Ji, Z. Zhou, H. Liu, and R. V. Davuluri. Dnabert: pre-trained bidirectional encoder representations from transformers model for dna-language in genome. *bioRxiv*, 2020.

N. Jiang and M.-C. de Marneffe. Evaluating bert for natural language inference: A case study on the commitmentbank. 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)*, pages 6088–6093, 2019.

D. Jin, Z. Jin, J. T. Zhou, and P. Szolovits. Is bert really robust? a strong baseline for natural language attack on text classification and entailment. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 8018–8025, 2020.

D. Jurafsky and J. H. Martin. Speech and language processing (draft). october 2019. *URL https://web. stanford. edu/˜ jurafsky/slp3*, 2019.

D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

P. R. Kingsbury and M. Palmer. From treebank to propbank. In *LREC*, pages 1989–1993. Citeseer, 2002.

P. Koehn. Statistical significance tests for machine translation evaluation. In *Proceedings of the 2004 conference on empirical methods in natural language processing*, pages 388–395, 2004.

J. R. Landis and G. G. Koch. The measurement of observer agreement for categorical data. *biometrics*, pages 159–174, 1977.

J. Lee, W. Yoon, S. Kim, D. Kim, S. Kim, C. H. So, and J. Kang. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240, 2020.

J. D. Lewis and J. L. Elman. Learnability and the statistical structure of language: Poverty of stimulus arguments revisited. In *Proceedings of the 26th annual Boston University conference on language development*, volume 1, pages 359–370. Citeseer, 2001.

P. Lewis, B. Oğuz, R. Rinott, S. Riedel, and H. Schwenk. Mlqa: Evaluating cross-lingual extractive question answering. *arXiv preprint arXiv:1910.07475*, 2019.

Z. Li, X. Ding, and T. Liu. Transbert: A three-stage pre-training technology for story-ending prediction. *ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP)*, 20(1):1–20, 2021.

Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

I. Loshchilov and F. Hutter. Decoupled weight decay regularization, 2019.

Z. Lu, P. Du, and J.-Y. Nie. Vgcn-bert: augmenting bert with graph embedding for text classification. In *European Conference on Information Retrieval*, pages 369–382. Springer, 2020.

B. C. Lust. *Child language: Acquisition and growth*. Cambridge University Press, 2006.

C. D. Manning. Computational linguistics and deep learning. *Computational Linguistics*, 41(4):701–707, 2015.

L. Martin, B. Muller, P. J. O. Suárez, Y. Dupont, L. Romary, É. V. de la Clergerie, D. Seddah, and B. Sagot. Camembert: a tasty french language model. *arXiv preprint arXiv:1911.03894*, 2019.

M. McShane. Natural language understanding (nlu, not nlp) in cognitive systems. *AI Magazine*, 38(4):43–56, 2017.

T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

W. Morgan. Statistical hypothesis tests for nlp, 2005.

B. Myagmar, J. Li, and S. Kimura. Transferable high-level representations of bert for cross-domain sentiment classification. In *Proceedings on the International Conference on Artificial Intelligence (ICAI)*, pages 135–141. The Steering Committee of The World Congress in Computer Science, Computer . . . , 2019.

M. Palmer, D. Gildea, and N. Xue. Semantic role labeling. *Synthesis Lectures on Human Language Technologies*, 3(1):1–103, 2010.

K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318, 2002.

M. Polignano, P. Basile, M. De Gemmis, G. Semeraro, and V. Basile. Alberto: Italian bert language understanding model for nlp challenging tasks based on tweets. In *6th Italian Conference on Computational Linguistics, CLiC-it 2019*, volume 2481, pages 1–6. CEUR, 2019.

L. A. Ramshaw and M. P. Marcus. Text chunking using transformation-based learning. In *Natural language processing using very large corpora*, pages 157–176. Springer, 1999.

A. Rogers, O. Kovaleva, and A. Rumshisky. A primer in bertology: What we know about how bert works, 2020.

J. R. Saffran, R. N. Aslin, and E. L. Newport. Statistical learning by 8-month-old infants. *Science*, 274(5294):1926–1928, 1996.

M. Sahlgren and F. Carlsson. The singleton fallacy: Why current critiques of language models miss the point. *arXiv preprint arXiv:2102.04310*, 2021.

T. Samardzic. *Dynamics, causation, duration in the predicate-argument structure of verbs: a computational approach based on parallel corpora.* PhD thesis, University of Geneva, 2013.

M. Sänger, U. Leser, S. Kemmerer, P. Adolphs, and R. Klinger. Scare—the sentiment corpus of app reviews with fine-grained annotations in german. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 1114–1121, 2016.

V. Sanh, L. Debut, J. Chaumond, and T. Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019.

K. R. Scherer and H. G. Wallbott. Evidence for universality and cultural variation of differential emotion response patterning. *Journal of personality and social psychology*, 66(2):310, 1994.

A. Schiller, S. Teufel, C. Stöckert, and C. Thielen. Guidelines für das tagging deutscher textcorpora. *University of Stuttgart/University of Tübingen*, 1999.

G. Schneider. *Hybrid long-distance functional dependency parsing.* PhD thesis, University of Zurich, 2008.

R. Schwartz, J. Dodge, N. A. Smith, and O. Etzioni. Green ai. *arXiv preprint arXiv:1907.10597*, 2019.

R. Sennrich, G. Schneider, M. Volk, and M. Warin. A new hybrid dependency parser for german. *Proceedings of the German Society for Computational Linguistics and Language Technology*, pages 115–124, 2009.

R. Sennrich, M. Volk, and G. Schneider. Exploiting synergies between open resources for german dependency parsing, pos-tagging, and morphological analysis. In *Proceedings of the International Conference Recent Advances in Natural Language Processing RANLP 2013*, pages 601–609, 2013.

O. Sharir, B. Peleg, and Y. Shoham. The cost of training nlp models: A concise overview. *arXiv preprint arXiv:2004.08900*, 2020.

R. Socher, A. Perelygin, J. Wu, J. Chuang, C. D. Manning, A. Y. Ng, and C. Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642, 2013.

C. Sun, A. Myers, C. Vondrick, K. Murphy, and C. Schmid. Videobert: A joint model for video and language representation learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7464–7473, 2019a.

Y. Sun, S. Wang, Y. Li, S. Feng, X. Chen, H. Zhang, X. Tian, D. Zhu, H. Tian, and H. Wu. Ernie: Enhanced representation through knowledge integration. *arXiv preprint arXiv:1904.09223*, 2019b.

E. Troiano, S. Padó, and R. Klinger. Crowdsourcing and validating event-focused emotion corpora for german and english. *arXiv preprint arXiv:1905.13618*, 2019.

A. Turing. Computing machinery and intelligence. *Mind*, 59(236):433, 1950.

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. *arXiv preprint arXiv:1706.03762*, 2017.

A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, and S. R. Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*, 2018.

A. Warstadt, A. Singh, and S. R. Bowman. Neural network acceptability judgments, 2019.

J. Welbl, N. F. Liu, and M. Gardner. Crowdsourcing multiple choice science questions, 2017.

A. Williams, N. Nangia, and S. R. Bowman. A broad-coverage challenge corpus for sentence understanding through inference. *arXiv preprint arXiv:1704.05426*, 2017.

E. Wittenberg. *With light verb constructions from syntax to concepts*, volume 7. Universitätsverlag Potsdam, 2016.

T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, et al. Transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*, 2019.

Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey, et al. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*, 2016.

Y. Yang, Y. Zhang, C. Tar, and J. Baldridge. Paws-x: A cross-lingual adversarial dataset for paraphrase identification. *arXiv preprint arXiv:1908.11828*, 2019.

A. Yeh. More accurate tests for the statistical significance of result differences. *arXiv preprint cs/0008005*, 2000.

Y. Zhang, J. Baldridge, and L. He. Paws: Paraphrase adversaries from word scrambling. *arXiv preprint arXiv:1904.01130*, 2019a.

Z. Zhang, Y. Wu, H. Zhao, Z. Li, S. Zhang, X. Zhou, and X. Zhou. Semantics-aware bert for language understanding. *arXiv preprint arXiv:1909.02209*, 2019b.

Y. Zhu, R. Kiros, R. Zemel, R. Salakhutdinov, R. Urtasun, A. Torralba, and S. Fidler. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *Proceedings of the IEEE international conference on computer vision*, pages 19–27, 2015.

Lebenslauf

Persönliche Angaben

Jonathan Schaber
Schartenstrasse 103
5430 Wettingen
jonathan.schaber@uzh.ch

Schulbildung

2006-2009	Fachmittelschule (FMS) Kantonsschule Wettingen
2009-2011	Matura Kantonsschule Wettingen
2012-2016	Bachelor-Studium Germanistik, Philosophie an der Universität Zürich
seit 2017	Master-Studium Computerlinguistik, historische Linguistik an der Universität Zürich

Berufliche und nebenberufliche Tätigkeiten

2012–2013	Tutorate PCL I+II

A Tables

Part of speech	POS type	POS	number of labels in my corpus
14	DET	**35**	280
14	DET	**35**	280
14	DET	**35**	280
14	DET	**35**	280
14	DET	**35**	280
14	DET	**35**	280
14	DET	**35**	280
14	DET	**35**	280
14	DET	**35**	280
14	DET	**35**	280
14	DET	**35**	280
14	DET	**35**	280
14	DET	**35**	280
14	DET	**35**	280
14	DET	**35**	280
14	DET	**35**	280
14	Total	**35**	280

Table 17: Some very large table in the appendix

B List of something

This appendix contains a list of things I used for my work.

- apples
  - `export2someformat`
- bananas
- oranges
  - `bleu4orange`
  - `rouge2orange`