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. 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 stated in an expression, and uphold such a state of affairs during the whole conversation. That this 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 tele-

scope".

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 communication. 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).[1] 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.[2]

[1] 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 [...]".

[2] 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

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 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 measure which performed better — as laid out before, this has led to the problematic temptation to focus on beating SOTAs without a great deal of theoretical linguistic considerations behind.

For almost 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. 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 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

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.

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. Several tasks have been formulized which model such processes, for example: NLI is the task of recpgnizing if sentence B is entailed by sentence B, or stands in a neutral or contradictory relationship to it. Question Answering is the task of — answering questions (see also section **??**. In sentiment detection, a model has to infer which emotion is expressed by an utterance.

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 acting 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.

1.1.2 Contextualized Word Embeddings in NLU

Since the beginning of the neural age, there was the problem as to how could text be numerically represented, so that the algorithms could extract meaningful feature patterns and that there is as little information loss as possible (however, since a numeric representation is always an abstraction of the real data, there naturally is some unpreventable information loss). The solution that was proposed by Mikolov et al. [2013] is the approach that is still in use today in its core idea:

- Initialize a random vector for each word in the vocabulary.
- Train a neural model to learn the best numerical representation of each word by giving it a simple task on huge amounts of unlabeled data (like CBOW, next word prediction, etc.).
- Save those numeric representations and use them in target task at hand.

While the basic approaches of this approach still hold — train randomly initialized vectors on large amounts of unlabeled data with a neural network with a simple training goal —, some important changes or additions to today's implementation have been made:

- The original word2vec embeddings were *fixed*, in the sense that a word had always the same representation, regardless of the context
- The neural networks that computed these vectors were quite small (two layers of dimensionality 300) and could be run on a standard machine. Today's models are huge (hundreds of millions of parameters are not unusual) and computationally very intensive and cannot be run locally.
- Due to the last point, practice has shifted towards pretraining these computationally heavy embeddings and finetuning them on the specific task along with it's goal

The architecture that has caused the most uproar 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.[3] Albeit this complete lack of any sort of

[3]Because of this non linguistic specific architecture, BERT can easily be adapted to operate on

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

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.

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

1.2 Research Questions

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

1. What do I do? Try to boost BERT embeddings on NLU tasks
2. How do I do it? Combine BERT embeddings with encoded SRL information, similar ti Zhang et al. [2019a]
3. And why? BERT lacks any linguistic knwoledge and shows error behaviour which suggests adding some could help
4. more concrete questions:
   - Can I reproduce Zhang et al. [2019b] for German?
   - Am I able to reach reported SOTAs of the data sets?

other sequential data. This has actually been done: Ji et al. [2020] for example trained a DNABERT model for successfully deciphering non-coding DNA.

  - Is there a difference for different head architectures? And if yes, why?
  - general outlooks

Questions and Claims that are **not** being made:

- No claim of "real" understanding
- carefully interpreting purely empirical results → data quality critique

1.3 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 rather trivial: (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 strengths of this approach is that the cost, hardware, and data intensive pre-training of the embeddings (step one) must only be computed once; the downstream task dependent fine-tunig can then be carried out in a lean set up.[2]

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

[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

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 optimized models. Thus, the NLP community was awestruck.

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.

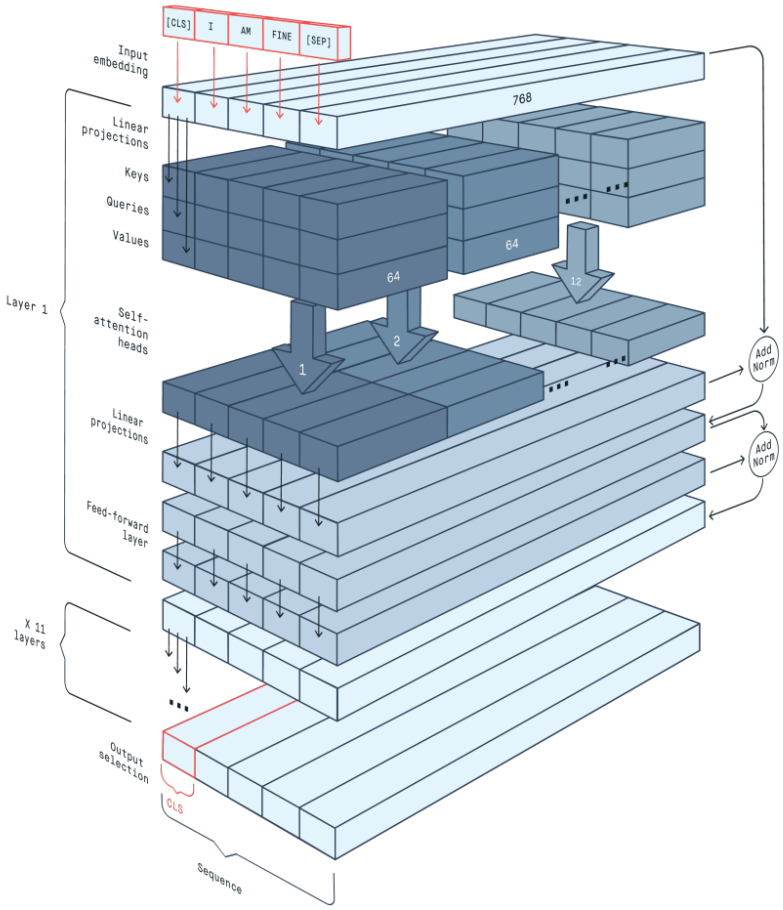

Figure 1: 3D-visualization of the BERT architecture. Nicely illustrated are the 12 attention heads and following linear projections in one block (here called "Layer"). Credit for the image goes to Peltarion

2.2 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.

2.3 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