Universität Zürich[UZH]

Masterarbeit
zur Erlangung des akademischen Grades
Master of Arts
der Philosophischen Fakultät der Universität Zürich

Leveraging Pretrained Word Embeddings by Enriching them with Linguistic Information During Fine-Tuning: A Case Study for germanBERT and Semantic Role Labels

Verfasserin/Verfasser: 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: 01.12.2020

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 X, 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
CPOSTAG	Coarse-grained Part-Of-Speech tag
GRU	Gated Recurrent Unit
LSTM	Long Short-Term Memory
ML	Machine Learning
NLP	Natural Language Processing
POS	Part-Of-Speech
POSTAG	Fine-grained Part-Of-Speech tag
RNN	Recurrent Neural Network
SRL	Semantic Role Labelling OR Semantic Role Labeller
STTS	Stuttgart-Tübingen-TagSet
USD	Universal Stanford Dependencies

1 Introduction

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. Another astonishing fact about human language is the overwhelming amount of languages that exist today, even that number was probably much higher a few centuries ago. As to how languages evolve, change over time and what trajectories of possible change may be, lots of questions are still open, and there remains enough work to do. But for me, maybe the most trivial and enigmatic trait about human language is that we actually *understand* each other: 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 a sentence, and uphold such a state of affairs during the whole conversation. That this is not as trivial as it might look like on first sight, show the following considerations: Human language is, when being used, notoriously ambiguous, metaphorical and formally corrupted.

So, every system that claims to process human language in a ... must be able

1.1.1 History, Methods, Problems of NLU

The subsection of NLP that deals with the semantics, i.e. meanings, of utterances, is NLU. For quite some time, as in most areas of NLP, systems that addressed NLU problems were architectures that consisted of carefully hand-written rules that aimed at tackling a specific problem, such as recognizing textual entailment, coreference resolution, sentiment analysis, and so on.

From on the 90ies, the so-called emphstatistical revolution took place, and NLU related problems were now being addressed by learning patterns from huge data collections. The main challenge for engineers and scientists now lay in dicovering suitable features, according to which the algorithm would hopefully learn helpful patterns for solving the task at hand.

Since now almost a decade, a next stage in NLU and NLP, in general, was entered — we are now deep in the neural age of computational linguistics. In contrast to the statistical period's main challenge, now the algorithm is even itself learning the features that are the most informative for a given task. The human part in the process is to design the overall model architecture and provide large enough amounts of data that are also of good quality.

In other words,

"The engineering side of computational linguistics, often called natural language processing (NLP), is largely concerned with building computational tools that do useful things with language" Johnson [2009]

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 meaningful represented, so that the algorithms can extract meaningful feature patterns and that there is as little information loss as possible (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]. These embeddings have also proved to achieve state-of-the-art results on well-established data sets, such as, e.g., GLUE Wang et al. [2018].

However, the many studies and experiments that have been carried out exploring the capabilities and mechanisms behind BERT quickly showed that nevertheless BERT performs on many tasks surprisingly well, even outperforming all models before it, there are situations, often trivial looking ones, where BERT desperately fails.

Jin et al. [2019], for example, showed that by creating adversarial examples in the test set — which were, of course, still valid —, they could bring down the performance of BERT by a large margin.

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.

One way to address the above outlined problems (1) the failure of very sophisticated models in rather trivial situations and (2) the difficulties of the interpretation of their output lies in bringing back again the linguistics into the whole picture.

Examples:

- Zhang et al SemBERT
- Goldberg Syntax to the rescue

- ...

1.2 Research Questions

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

1. What do I do?
2. How do I do it?
3. And why?
4. Can I reproduce Zhang et al. [2019b] for German?
5. Am I able to reach reported SOTAs of the data sets?
6. Is there a difference for different head architectures? And if yes, why?

1.3 Thesis Structure

In this first chapter . . .
Chapter 2 introduces . . .
Chapter 3 . . .

2 Semantic Roles

2.1 Overview

"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]

3 Data Sets

3.1 gliGLUE

Traditionally in linguistics, language is analyzed into different structural levels, where different tools for describing these levels, or strata, are used. In most theories, there are are four of these structural levels proposed: Beginning from the Bottom, there is the level of Phonetics and Phonology, followed by Morphology, then there is the level of Syntax, and the last one is Semantics.[0] While the first three levels deal with the form of utterances of human language, semantics is concerned with the meaning of such utterances [Kracht, 2007, p. 4ff.].

Following Wang et al. [2018],

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
		Two-Sentence 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[0]	105k/5.4k	train/test
QQP	Paraphrase Identification	Binary Classification	364k/391k	train/test
RTE	Natural Language Inference	Binary Classification[0]	2.5k/3k	train/test
STS-B	Sentence Similarity	Regression (1 - 5)	7k/1.4k	train/test
WNLI	Coreference Resolution	Binary Classification[0]	634/146	train/test

Table 1: Original GLUE data sets and tasks.

[0] Sometimes Pragmatics is conceptualized as an additional fifth layer on top, sometimes it is considered to form a field of its own; I follow the latter.

[0] 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.

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	-
		Two-Sentence Tasks		
MLQA	Question Answering	Span Prediction	509/4 499	dev/test
PAWS-X	Paraphrase Identification	Binary Classification	14 402/2 000/4 000	train/dev/test
XNLI	Natural Language Inference	Multi-Class Classification	2 489/7 498	dev/test
XQuAD	Question Answering	Span Prediction	1 192	-

Table 2: gliGLUE data sets and tasks.

3.1.1 General Issues

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

(1) Most of the data sets 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.

[0] 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*.

[0] 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.

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, classifiaction 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 exmample, 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) Most datasets that I collected are not mono- but multilingual compilations, in which case I simply include the German subset of the corpus into my GLUE.

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, how looks the label distribution, etc.

3.2.1 deISEAR

3.2.1.1 Task

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, predict the correct label from a fixed set of emotions. Following by 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, so the annotators, which were recruited via an crowdsourcing platform, had to complete sentences having the following structure: "Ich fühlte *emotion*, als/weil...". Seven emotions were given for which the descriptions had to be constructed: Traurigkeit, Ekel, Schuld, Wut, Angst, Scham, Freude. For *Traurigkeit* and *Ekel* there are 144 examples in the data set, for the other emotions there are 143.

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

The searched emotion is *Traurigkeit* in example 3.2.1.1.

3.2.1.2 Statistics

number of examples:
train: 700
dev: 150
test: 151

merged
average length train: 15.9 (sigma 6.6)
average length dev: 17.9 (sigma 19.9)
average length test: 17.1 (sigma 7.4)

subtokenized
average length train: 18.1 (sigma 7.9)
average length dev: 20.7 (sigma 24.1)
average length test: 19.5 (sigma 8.8)

3.2.1.3 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.

3.2.2 MLQA

3.2.2.1 Task

(3.2) 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.

1. Wann wurde Rita Sahatçiu Ora geboren? → 26. November 1990

Lewis et al. [2019] compiled

PROBLEM: 231 out of 5,008 exceed tokenized length of 512 → ignore? 4.6%

3.2.2.2 Statistics

number of examples:
train: 432
dev: 77
test: 4,499

merged
average length train answer: 4.0 (sigma 4.9)
average length dev answer: 3.7 (sigma 5.4)
average length test answer: 4.0 (sigma 5.1)

average length train question: 9.4 (sigma 3.7)
average length dev question: 8.6 (sigma 3.4)
average length test question: 9.1 (sigma 3.4)

average length train context: 127.7 (sigma 110.0)
average length dev context: 125.1 (sigma 116.7)
average length test context: 129.9 (sigma 123.1)

subtokenized
average length train answer: 5.6 (sigma 6.6)
average length dev answer: 5.2 (sigma 6.7)
average length test answer: 5.6 (sigma 7.0)

average length train question: 11.4 (sigma 4.5)
average length dev question: 10.6 (sigma 4.3)
average length test question: 11.2 (sigma 4.3)

average length train context: 162.7 (sigma 139.0)
average length dev context: 159.4 (sigma 145.6)
average length test context: 165.5 (sigma 156.7)

3.2.2.3 SOTA

Lewis et al. [2019] train their cross-lingual transfer model on the 100,000 instances of SQuAD Rajpurkar et al. [2016] as training data. They use the English development set of MLQA for tuning. At test time, the model must extract the answer span in the target language. They report that XLM performs best for German, achieving a 47.6% accuracy of exact matches, i.e. predicting the correct start and end span of the answer.

The total of all instances in all languages in MLQA is 46,444.

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].

(3.3) 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.

The label for the sentence pair 3.2.3, of course, would be *true*, since sentence one is a paraphrase of sentence two, and vice versa.

stats

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.[0] Further, some examples consist of empty strings; I treat those the same way as the examples mentioned before.

Further, there are sentences XXXXX

3.2.3.2 Statistics

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.

Train: 0.553
Development: 0.373
Test: 0.384

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

merged
average lenght sentence 1 train: 21.0 (sigma 6.5)
average length sentence 2 train: 21.0 (sigma 5.8)

average length sentence 1 dev: 21.1 (sigma 6.0)
average length sentence 2 dev: 21.1 (sigma 6.0)

average length sentence 1 test: 21.4 (sigma 5.9)
average length sentence 2 test: 21.3 (sigma 5.9)

subtokenized
average length sentence 1 train: 27.5 (sigma 9.0)
average length sentence 2 train: 27.4 (sigma 8.2)

average length sentence 1 dev: 27.6 (sigma 8.4)
average length sentence 2 dev: 27.7 (sigma 8.4)

[0] 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.

average length sentence 1 test: 28.1 (sigma 8.4)
average length sentence 2 test: 28.0 (sigma 8.4)

The training set contains 3,209 sentence pairs (6.6% of all the sentence pairs) with a BLEU score of 1.0 — which means they are identical.

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.

3.2.4 SCARE

3.2.4.1 SCARE normal

"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:

(3.4) guter wecker... || vom prinzip her echt gut...aber grade was die sprachausgabe betrifft noch etwas buggy....[0]

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

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.

Example from .csv 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 3: An example from the alarm_clocks.csv file.

Corresponding .rel file:

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

[0]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.

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 4: An example from the alarm_clocks.rel file.

3.2.4.2 SCARE reviews

Besides their carefully, hand-annotated corpus, the authors also provide a dataset comprising of 802,860 reviews along with the rating — one to five stars —, that were available in German on the Google Play Store. This data set is much larger than the annotated one: Due to the great expenses of generating those fine-grained annotations, the authors were able to annotate only 0.22% of all reviews available.

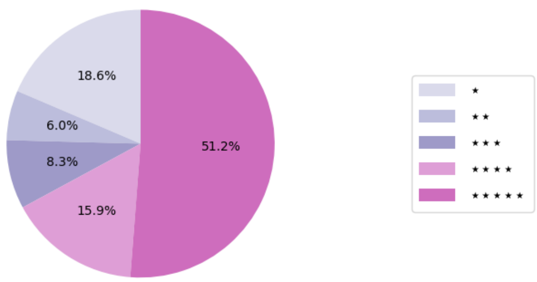

Figure 1: Overview of percentage of stars given. Clearly, there is an imbalance towards giving the full amount of stars possible

3.2.4.3 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-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:

(3.5) “Ich bin die erfuinderin || Ich bin die erfunden!!!!!!!!!!!!!!!!!!!!!!!!!!”.

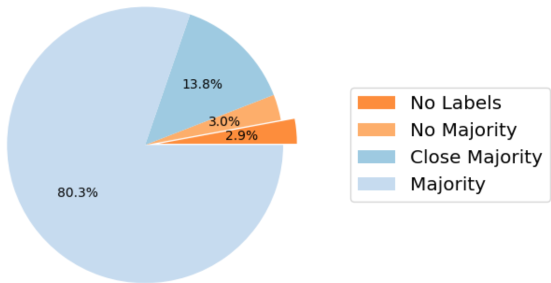

Figure 2: Statistics of label generation. For most of the examples, there was a clear majority decision as to which label should be chosen. *Close Majority* means the majority vote was off by 1. The reddish portions in the graph were labelled *neutral* by default, while the blueish ones were labelled accoring to the majority vote decision.

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)

3.2.4.4 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)

3.2.4.5 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 aspect and 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] build 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.

(3.6) 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.

The label for example 3.2.5 is *neutral* since the second sentence does not follow necessarily from the first and it also does not contradict it, either.

3.2.5.1 Statistics

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

merged

subtokenized

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 accuracy 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].

(3.7) 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.

The questions and corresponding answer spans for paragraph 3.2.6 in the data set are the following:

1. Wer leitete eine philosophische Diskussion über Kraft? → Aristoteles
2. Wovon war das Konzept der Kraft ein integraler Bestandteil? → aristotelischen Kosmologie
3. Aus wie vielen Elementen besteht die irdische Sphäre nach Ansicht des Aristoteles? → vier
4. Wo vermutete Aristoteles den natürlichen Ort für Erd- und Wasserelemente? → auf dem Boden
5. Was bezeichnete Aristoteles als erzwungene Bewegung? → unnatürlichen

Artetxe et al. [2019]

3.2.6.1 Statistics

stats:

average length train answer: 3.2 (4.7)
average length dev answer: 3.3 (5.2)
average length test answer: 3.6 (5.7)

average length train question: 11.3 (14.3)
average length dev question: 11.5 (14.3)
average length test question: 11.4 (14.5)

average length train context: 151.3 (191.7)
average length dev context: 149.5 (190.7)
average length test context: 144.3 (187.3)

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 monlingual BERT in English with original pretraining objectives
2. Transfer model to new language L_2, but learn only token embeddings new (transformer body is freezed) 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

F1: 73.6 Accuracy (exact match): 57.6%

3.2.7 Overview

Data Set	NLP Task	ML Task	# Examples	Splits
deISEAR	Emotion Detection	Sequence Classification	XYZ	-
MLQA	Question Answering	Span Prediction	XYZ	dev/test
PAWS-X	Paraphrase Identification	Sequence Classification	XYZ	train/dev/test
SCARE	Sentiment Analysis	Sequence Classifiaction	XYZ	-
SCARE Rev.	Sentiment Analysis	Sequence Classification	XYZ	-
XNLI	Natural Language Inference	Sequence Classification	XYZ	dev/test
XQuAD	Question Answering	Span Prediction	XYZ	-

Table 5: Overview of collected data sets and tasks.

4 Architecture

4.1 Overview

4.2 Semantic Role Labeller

A Semantic Role Labeller (SRL) is a system, that assigns automatically semantic roles to a given input text.[0]

State-of-the-art semantic role labellers (SRLs) are end-to-end models, nowadays often implementing deep learning techniques, like RNNs or attention, 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. Since one sentence can have multiple predicate-argument structures, I need to device an algorithm to extract the relevant predicates in a sentence. This is not as straight forward as it seems on the first look.

4.2.1 Finding Predicates

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

[0] This may be one or multiple sentences.

"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 USD [De Marneffe et al., 2014], which explicitly sets the content verb as root, identification of the relevant predicate is straight-forward: One has simply to look at the dependency parse tree of a given sentence and select the heads — i.e. roots — of the clauses. However, the ParZu parser models not content words as heads but function words.[0]

(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.", *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 3: This sentence has five verbs in it, *wurde*, *abgewiesen*, *beschrieben*, *werden*, and *kann* (POS-tag "V" in the second

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

row), but only two of them are relevant predicates, i.e. predicates that carry "true" semantics.

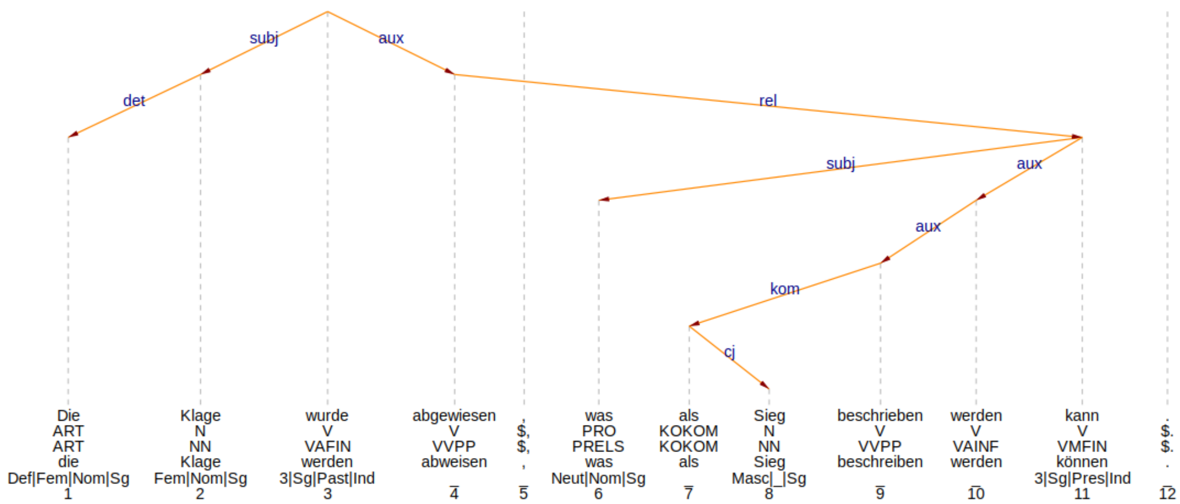

Figure 3: Example dependency parse tree for a sentence with multiple predicates.

I propose the following algorithm 1 deciding whether a verb in a sentence is or isn't a predicate using a heuristic, relying on the token's POS tag that the parser 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 4.2.1: 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 4.2.1 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

```
for all token t ∈ sentence do
    if CPOSTAG t ≠ 'V' then
        t ← NOT_PRED
    else
        if POSTAG t = 'VVFIN' then
            t ← PRED
        else
            FLAG ← True
            for all token u ≠ t ∈ subclause where t ∈ subclause do
                if CPOSTAG u = 'V' ∧ u dependent on t then
                    t ← NOT_PRED
                    FLAG ← False
                    break
                end if
            end for
            if FLAG = True then
                t ← PRED
            end if
        end if
    end if
end for
```

4.2.2 DAMESRL

4.3 German BERT

Since its publishing two years ago, BERT [Devlin et al., 2018] has often been called a "turning-point" in ML in NLP.

I use the `bert-base-german-cased` model from deepset which is available in pyTorch through the hugging face libraryWolf et al. [2019].

4.3.1 Combining BERT encodings and SRLs

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 theflexibility 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:

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

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

(4.6) 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.

4.3.2 Final Layer

4.3.2.1 Question Answering

4.3.2.2 Classifiaction

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.

5 Results

5.1 SRL Evaluation

5.1.1 delSEAR

5.1.1.1 Example 1

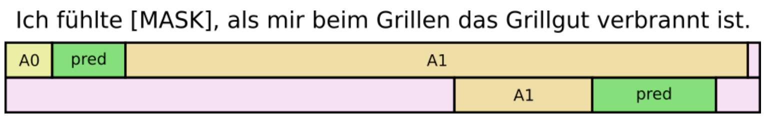

Figure 4:

5.1.1.2 Example 2

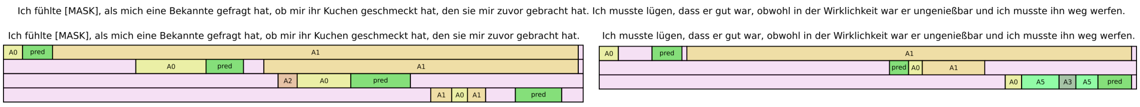

Figure 5:

5.1.2 PAWS-X

5.1.2.1 Example 1

Sentence 1

Im Gegenzug [$_{\text{predicate}}$ gab] Grimoald [$_{\text{A1}}$ seine Tochter zur Hochzeit] und gewährte ihm das Herzogtum Spoleto nach dem Tod von Atto.

Im Gegenzug gab Grimoald [$_{\text{A0}}$ seine Tochter] zur Hochzeit und [$_{\text{predicate}}$ gewährte] [$_{\text{A2}}$ ihm] [$_{\text{A1}}$ das Herzogtum Spoleto nach dem Tod von Atto] .

Sentence 2

Im Gegenzug [$_{\text{predicate}}$ gab] Grimoald [$_{\text{A1}}$ seine Tochter] [$_{\text{A3}}$ in die Ehe] und gewährte ihm das Herzogtum Spoleto nach dem Tod von Atto.

Im Gegenzug gab Grimoald [$_{\text{A0}}$ seine Tochter] in die Ehe und [$_{\text{predicate}}$ gewährte] [$_{\text{A2}}$ ihm] [$_{\text{A1}}$ das Herzogtum Spoleto nach dem Tod von Atto] .

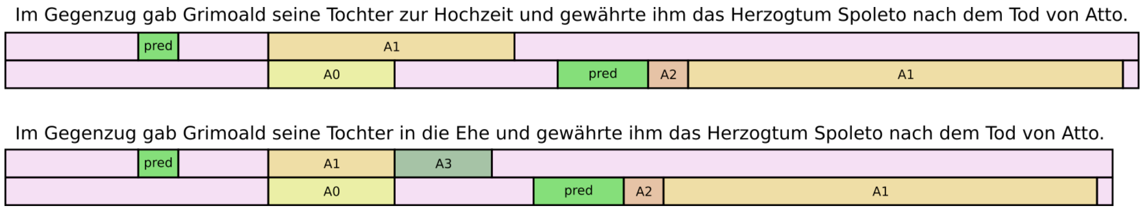

Figure 6:

5.1.2.2 Example 2

Sentence 1

Camm $[_{\text{predicate}}$ entschied] , $[_{\text{A1}}$ dass beide Motoren eingesetzt werden sollten: Der Tempest Mk 5 hatte den Napier Saber eingebaut, während der Tempest Mk 2 der Bristol Centaurus war] .

Camm entschied, dass $[_{\text{A1}}$ beide Motoren] $[_{\text{predicate}}$ eingesetzt] werden sollten: $[_{\text{A1}}$ Der Tempest Mk 5 hatte den Napier Saber eingebaut, während] der Tempest Mk 2 der Bristol Centaurus war.

Camm entschied, dass beide Motoren eingesetzt werden sollten: $[_{\text{A0}}$ Der Tempest Mk 5] hatte $[_{\text{A3}}$ den Napier Saber] $[_{\text{predicate}}$ eingebaut], während der Tempest Mk 2 der Bristol Centaurus war.

Camm entschied, dass beide Motoren eingesetzt werden sollten: Der Tempest Mk 5 hatte den Napier Saber eingebaut, während $[_{\text{A1}}$ der Tempest Mk 2 der Bristol Centaurus] $[_{\text{predicate}}$ war] .

Sentence 2

Camm $[_{\text{predicate}}$ entschied] , $[_{\text{A1}}$ dass beide Motoren eingesetz werden sollten: Der Tempest Mk 5 war mit dem Napier Saber ausgestattet, während der Tempest Mk 2 den Bristol Centaurus hatte] .

Camm entschied, dass $[_{\text{A1}}$ beide Motoren] $[_{\text{predicate}}$ eingesetzt] werden sollten: $[_{\text{A1}}$ Der Tempest Mk 5 war mit dem Napier Saber ausgestattet, während der Tempest Mk 2 den Bristol Centaurus hatte] .

Camm entschied, dass beide Motoren eingesetzt werden sollten: $[_{\text{A0}}$ Der Tempest Mk 5] war $[_{\text{A1}}$ mit dem Napier Saber] $[_{\text{predicate}}$ ausgestattet] , während der Tempest Mk 2 den Bristol Centaurus hatte.

Camm entschied, dass beide Motoren eingesetzt werden sollten: Der Tempest Mk 5 war mit dem Napier Saber ausgestattet, während $[_{\text{A1}}$ der Tempest Mk 2 den Bristol Centaurus] $[_{\text{predicate}}$ hatte] .

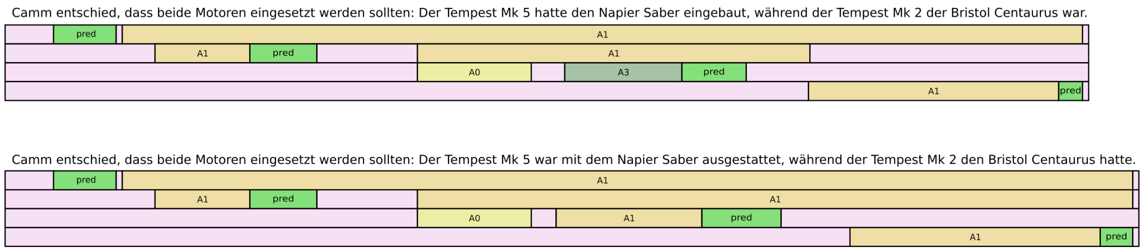
Camm entschied, dass beide Motoren eingesetzt werden sollten: Der Tempest Mk 5 hatte den Napier Saber eingebaut, während der Tempest Mk 2 der Bristol Centaurus war.
pred
A1
A1
pred
A1
A0
A3
pred
A1
pred
Camm entschied, dass beide Motoren eingesetzt werden sollten: Der Tempest Mk 5 war mit dem Napier Saber ausgestattet, während der Tempest Mk 2 den Bristol Centaurus hatte.
pred
A1
A1
pred
A1
A0
A1
pred
A1
pred

Figure 7:

5.1.2.3 Example 3

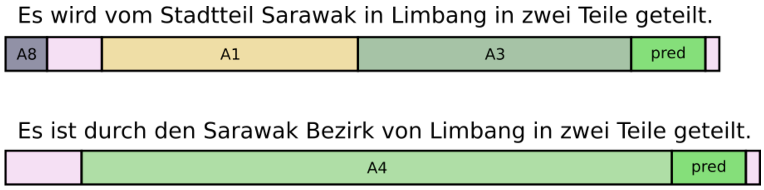

Figure 8:

5.1.2.4 Example 4

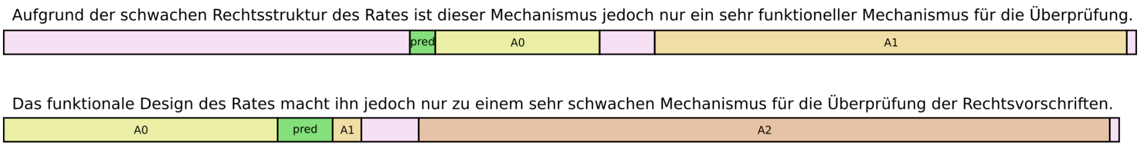

Figure 9:

5.1.2.5 Example 5

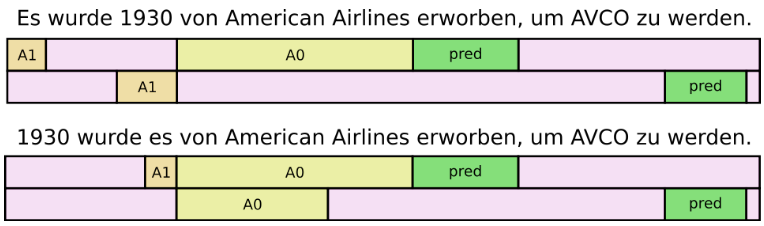

Figure 10:

5.1.2.6 Example 6

5.1.2.7 Example 7

5.1.2.8 Example 8

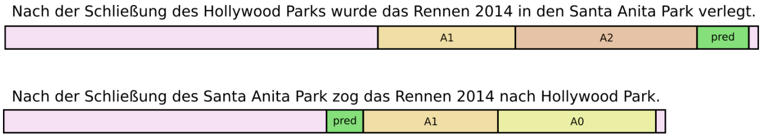

Figure 11:

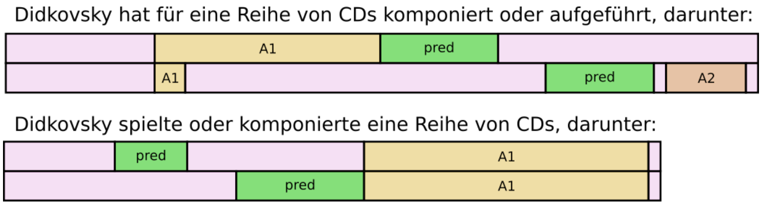

Figure 12:

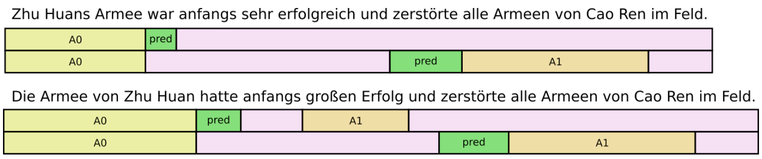

Figure 13:

6 Conclusion

In this project we have done so much.[1]

We could show that ...

Future research is needed.
The show must go on.

[1] Thanks to many people that helped me.

Glossary

Of course there are plenty of glossaries out there! One (not too serious) example is the online MT glossary of Kevin Knight [2] 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.

[2] 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. 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.

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.

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.

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

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.

D. Jin, Z. Jin, J. T. Zhou, and P. Szolovits. Is bert really robust? natural language attack on text classification and entailment. *arXiv preprint arXiv:1907.11932*, 2, 2019.

M. Johnson. How the statistical revolution changes (computational) linguistics. In *Proceedings of the EACL 2009 Workshop on the Interaction between Linguistics and Computational Linguistics: Virtuous, Vicious or Vacuous?*, pages 3–11, 2009.

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

M. Kracht. Introduction to linguistics. *Departement of*, 2007.

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

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.

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.

P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*, 2016.

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

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.

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. 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.

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.

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. 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.

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.

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.

Lebenslauf

Persönliche Angaben

Ich Persönlich
Meinestrasse Nr
PLZ Wohnort
ichpersoenlich@uzh.ch

Schulbildung

2012-2014	Bachelor-Studium Computerlinguistik und Sprachtechnologie an der Universität Zürich
seit 2014	Master

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	Total	**35**	280

Table 6: 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`