Argument Component Identification im Stile von Stab und Gurevych

Hugo Meinhof, 815220

March 1, 2024

Abstract

some abstract stuff goes here Stab und Gurevych

1 indtroduction

1.1 Previous Work: Stab and Gurevych 2017

In 2017, Stab and Gurevych revisited their own work on argumentation mining from 2014. Dissatisfied with the existing corpora at the time, they decided to make their own, which they extended in 2017. On that corpus of persuasive essays, they annotated argument components and their argumentation structures. on these annotations, they trained models to built and tested a pipeline. "However, our identification model yields good accuracy and an α_U of 0.958 for identifying argument components. Therefore, it is unlikely that identification errors will significantly influence the outcome of the downstream models when applied to persuasive essays." meaning that they didnt evaluate the entire pipeline, as they expect to get the same results as the models, running on gold data. This is a questionable practice, expecially considering that they're focussing on the accuracy, and not f1 here. cant find the accuracy referenced here "However, as demonstrated by Levy et al. (2014) and Goudas et al. (2014), the identification of argument components is more complex in other text genres than it is in persuasive essays. Another potential issue of the pipeline architecture is that wrongly classified major claims will decrease the accuracy of the model because they are not integrated in the joint modeling approach. For this reason, it is worthwhile to experiment in future work with structured machine learning methods that incorporate several tasks in one model (Moens 2013)." I have trained such model, full_labels, which sadly doesnt reach the f1 of the SuG ILP model yet.

1.2 Goal of my work

For this paper I have attempted to surpass the models for argument component identification and classification, as shown in Figure 1. In general, the identification reagards the question of where an argument component, like MajorClaim, will be located, without knowing what type it will be. Then, the classification task is the labeling of what kind of argument component we are dealing with. Stab and Gurevych trained models for each task, and my goal was to surpass them. ELABORATE AND REFERENCE FIGURE

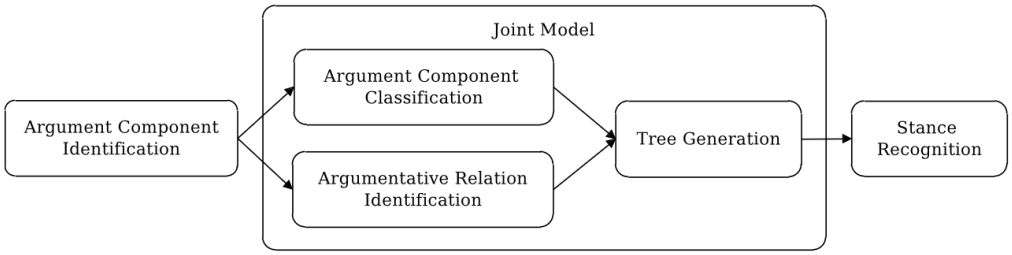

Figure 1: Architecture of the argumentation structure parser.

2 trainingsdaten . . . . . .

The foundation of the training data for this project consists of 402 student essays annotated by Stab and Gurevych. For each essay, the dataset includes information on where the core assertion of the text, otherwise known as the MajorClaim, supporting assertments, or the Claims, and the premises, that back up the thesis with statistics and other evidence, are located. The texts used were sourced from essayforum.com and selected randomly. No further information is known about the authors, but it is conceivable that English may not be their first language and they are likely learning it in an educational context. Stab and Gurevych subsequently annotated the essays with tokens and recorded the locations of spans.

quelle: https://aclanthology.org/J17-3005.pdf

Claim: semantische klasse, die eine behauptung aufstellt; stützt majorclaim MajorClaim: kernbehauptung(en) des textes prämissen: unterstützung/ untermauerung der Claims (zb statistiken)

über baselines schreiben!

"For finding the best-performing models, we conduct model selection on our training data using 5-fold cross-validation"

die essays wurden tokenisiert und gespeichert wo sich spannen befinden, welche rolle sie haben, und welche tokens dazugehören. diese daten werden von renes script erstellt was war das. damit modeelle damit lernen können, muss ein dataset erstellt werden, welches die daten aufbereitet und dem modell verständlich sortiert. dies ist die größte aufgabe beim training. da alle traininerten modelle auf dem selben datensatz basieren, gibt es für alle zusammen ein gemeinsames dataset. viele extraktions, aufbereitungs, und matching schritte bleiben für alle modelle gleich. unterschiede gibt es im grunde nur im letzten aufbereitungsschritt. das wird von den verschiedenen configs des datasets gehandhabt. es gibt für jedes modell eine eigene config, die den selben namen trägt, wie das modell. da sich die modelle nur darin unterscheiden wie die trainingsdaten aufbereitet sind, bedeutet das auch, dass ein trainingsscript für alle modelle verwendet werden kann, in dem nur die config angepasst werden muss. ich habe zudem darauf geachtet, dass die configs die selben namen tragen wie die modelle, damit alles reibungslos abläuft bessere erklärung des trainings scripts. das war jedoch nicht schon immer so. angefangen habe ich mit je einem trainings script pro modell. das ist zwar auf der einen seite nicht so anpassbar wie ein sript für alle, welches über command line arguments angepasst werden kann, hat jedoch auf der anderen seite den klaren vorteil, dass so ein einzelnes modell erstmal trainiert und ausgetestet werden kann, ohne dabei andere im hinterkopf behalten zu müssen.

The corpus consists of 402 essays, downloaded, and randomly selected, from essayforum.com.

3 evaluation

Makro-f1	full labels	spans	simple	sep tok full labels	sep tok
4	0.579	0.898	0.769	0.749	0.843
5	0.631	0.905	0.782	0.801	0.849
6	0.716	0.908	0.790	0.816	0.858
7	0.740	0.910	0.796	0.824	0.864
8	0.752	0.911	0.800	0.832	0.864
9	0.757	0.912	0.801	0.837	0.865
10	0.759	0.912	0.801	0.838	0.867

Table 1: 5-fold cross-validation of the macro-f1

Argument Component Identification	Makro-f1
Stab und Gurevytch	
Human upper bound	0.886
Baseline majority	0.259
Baseline heuristic	0.628
CRF all features	**0.849**
Meinhof	
spans	**0.912**

Table 2: Argument Component Identification (5-fold cross-validation as in Table C.1)

Argument Component Classification	Makro-f1
Stab und Gurevych	
Baseline majority	0.257
Baseline heuristic	0.724
SVM all features	**0.773**
ILP-balanced	**0.823**
ILP and CRF	no-eval
Meinhof	
full_labels	0.759
simple	0.801
sep_tok_full_labels	0.838
sep_tok	**0.867**
full_pipe	*TO-DO*

Table 3: Argument Component Classification (5-fold cross-validation as in Table C.2)

4 training