Gorynych Transformer at SemEval-2020 Task 6: Multi-task Learning for Definition Extraction

Adis Davletov
RANEPA
Lomonosov Moscow State University
davletov-aa@ranepa.ru

Nikolay Arefyev
Lomonosov Moscow State University,
Samsung R&D Institute Russia,
National Research University
Higher School of Economics
nick.arefyev@gmail.com

Alexander Shatilov, Denis Gordeev and **Alexey Rey**
RANEPA
shatilov-aa, gordeev-di, rey-ai@ranepa.ru

Abstract

This paper describes our approach to "DeftEval: Extracting Definitions from Free Text in Textbooks" competition held as a part of SemEval 2020. The task was devoted to finding and labeling definitions in texts. DeftEval was split into three subtasks: sentence classification, sequence labeling and relation classification. Our solution ranked 5th in the first subtask and 23rd and 21st in the second and the third subtasks respectively. Our best solution for subtasks 1, 3 employs multi-task learning of a Transformer-based model on all three tasks. However, for subtask 2 single-task learning proved to perform better.

1 Introduction

This work is devoted to DeftEval challenge (Spala et al., 2020) held as part of SemEval 2020. It was concerned with the problem of definition extraction. It has recently been a popular topic. However, there were few annotated datasets and they were often small in size (Jin et al., 2013) or were limited to the cases when a term and its definition are in the same sentence (Navigli et al., 2010).

DeftEval is one of the first attempts to provide a structured multi-task dataset that can be used for various tasks connected with definition extraction and labeling (Spala et al., 2019) at text level (in contrast to sentence level). All the provided data was in English.

Our system employs a Transformer-based model trained jointly for all three tasks. For each task, we add a linear layer with dropout on top of the Transformer output. It allows the system to use information about sentence classes, entities it contains, and relations between them at the same time while training. This helps to improve the results for Subtasks 1,3. However, a single-task model performs better for Subtask 2.

Our system achieved F1 score of 0.844 for the first task with the difference from the first place equal to approximately 0.03 points. For the second task, our final score was equal to 0.52 while the difference amounted to 0.32. For the third task, our F1-score was equal to 0.61 while the winning system achieved the perfect score of 1.0. Although named entity information was provided for the third subtask, we did not use it and included only span information into the model. The main contribution of our work is a detailed analysis of multi-task systems performance for definition extraction, classification and named entity recognition. The results show that our approach to multi-task training might be beneficial for the sequence classification task, but it requires reconsidering for sequence labeling. Our code is publicly available[1].

2 Background

DeftEval was split into three subtasks [2]:

[1] Gorynych is a Russian folklore three-headed dragon. https://github.com/davletov-aa/deft-eval-2020

[2] https://competitions.codalab.org/competitions/22759

Proceedings of the 14th International Workshop on Semantic Evaluation, pages 487–493
Barcelona, Spain (Online), December 12, 2020.

- Subtask 1: Sentence Classification
  Given a sentence, classify whether or not it contains a definition. This is the traditional definition extraction task.

- Subtask 2: Sequence Labeling
  Label each token with BIO tags according to the corpus' tag specification.

- Subtask 3: Relation Classification
  Given the tag sequence labels, label the relations between each tag according to the corpus' relation specification.

The dataset contained 215 files, 80 out of them were for training, 68 for validation, and 67 for the test. These files contained 7001 text extracts with 26552 sentences and 513219 word uses. The dataset contains 29011 unique tokens. The data was provided in CoNLL 2003-like format (see Fig. 1). The data was from several distinct domains: biology, economics, government, history, physics, psychology and sociology.

DeftEval was held in two phases: first, there was given the data for the first two subtasks which did not contain named entity information. Then the third subtask data with named entity spans and types were revealed.

```
The         data/source_txt/t7_government_0_404.deft        476     479     O       -1      -1      0
U.S.        data/source_txt/t7_government_0_404.deft        480     484     O       -1      -1      0
Constitution        data/source_txt/t7_government_0_404.deft        485     497     O       -1      -1      0
outlines            data/source_txt/t7_government_0_404.deft        498     506     O       -1      -1      0
the         data/source_txt/t7_government_0_404.deft        507     510     O       -1      -1      0
treaty      data/source_txt/t7_government_0_404.deft        511     517     O       -1      -1      0
process     data/source_txt/t7_government_0_404.deft        518     525     O       -1      -1      0
in          data/source_txt/t7_government_0_404.deft        526     528     O       -1      -1      0
Article     data/source_txt/t7_government_0_404.deft        529     536     O       -1      -1      0
II          data/source_txt/t7_government_0_404.deft        537     539     O       -1      -1      0
.           data/source_txt/t7_government_0_404.deft        539     540     O       -1      -1      0
```

Figure 1: Deft corpus example

There are many ways to extract information from text. This task is often solved by extracting named entities and classifying relations between them. Currently, the best results are achieved with Transformer-based models (Vaswani et al., 2017). The most advanced models (according to paperswithcode[3]) use extra training data or additional knowledge bases. For example, in the state-of-the-art system the authors use Wikipedia data (Baldini Soares et al., 2019). However, such data is impossible to get for domain-specific relations.

Among the systems that do not use encyclopedias or other labeled data, the best results were achieved by Joshi et al. (Joshi et al., 2019). They pre-trained a BERT-like system, but instead of predicting individual masked tokens, they trained the model to infer contiguous random spans. The model was also trained to predict each token in the masked span using output representations of only span boundary tokens. This significantly improved the results of their model in comparison with the vanilla BERT.

Our system applies a sequence labeling approach to both named entity recognition and relation extraction. A similar work was proposed by Veyseh et al. (Veyseh et al., 2020) where they built a joint system for definition extraction where they combined both sentence classification and sequence labelling in a single BiLSTM model with a graph convolutional layer on top of it. This approach looked promising and we decided to transform it and to use a single BERT-based model. Thus, we adopt a multi-task approach and predict sentence classes, named entities and the relation between these entities in one go. We compare the multi-task model results with its single-task counterparts.

3 System Overview

3.1 Multi-task learning

To solve all three subtasks of the competition, we decided to use the joint training method. We propose a model that simultaneously predicts an input example class, a tag sequence of entity labels and semantic

[3]https://paperswithcode.com/sota/relation-extraction-on-tacred

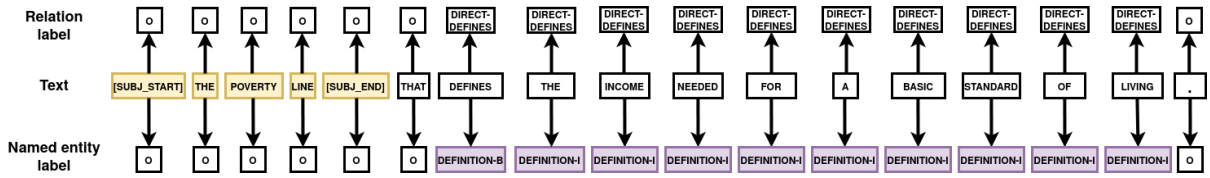

Figure 2: Joint relation extraction and named entity recognition

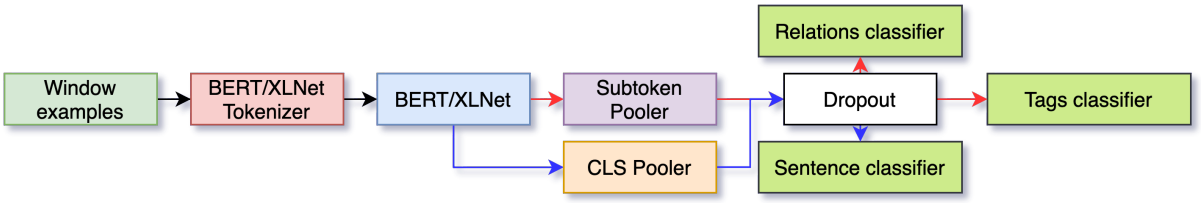

Figure 3: Architecture of our main model for joint subtasks learning.

relations between entities. To do so, we consider relation extraction as a sequence labeling problem (similar to how named entity recognition is usually solved). In each example, we have one marked main entity (which may contain several tokens) and we predict all named entity tags and all relations between the main entity and all other tokens in the sentence (see Fig. 2). The architecture of our main model is depicted in Figure 3.

The dataset contained texts separated into small windows of 3-5 sentences each. Windows were split with respect to their description ids. According to the organizers, there were no relations that span across windows. Thus, all our training and inference was done with respect to these windows.

In each training example, we highlight the boundaries of the analyzed sentence with special tokens.

We also mark the boundaries of the entity for which we are going to predict all relations in the text extract. So for each named entity from the dataset we generate a training example containing the boundaries for the considered entity and the sentence.

During training, the weighted sum of cross-entropies of each subtask was optimized. Thus, the proposed model relies solely on the input text and the knowledge of the boundaries of entities and sentences, without using information about entity types. The learning rate was set to 1e-5, the weight decay and the dropout were set to 0.1. To obtain information about entity boundaries, we trained an independent entity extraction model based on BERT (Devlin et al., 2018). We use it to extract entities and generate examples for our main model. In the competition for the third subtask, we used annotated named entity information provided by the organizers instead of BERT-model predictions.

BERT and XLNet (Yang et al., 2019) tokenizers split tokens into several subtokens so we had to create an aggregation scheme to merge subtoken outputs back together for entity and relation inference. The output was taken from the first sub-token.

For each training example from the training dataset we generated several samples. Since each example from the training dataset was turned into several examples, in the prediction we had to choose the answer from one. For this, in the first task, we selected the answer with the maximum score, for the second task for each word we took the answer from the example in which the score was maximum for this word. In the third task, we group examples into non-overlapping sets according to the predicted relation type, and in each set, we choose an example with the maximum average score. For the final prediction, we merge the answers of these sets.

Similar to the approach described before, we tried to train models jointly on first and seconds subtasks only. We also tried single-task models for the first and second tasks. Subtask sections 1 and 2 below describe the response construction process for each subtask respectively.

3.2 Single-task models

3.2.1 Subtask 1. Sentence classification

In this experiment, sentences were classified into two classes whether they contain a definition or not. As a single task model we fine-tuned a Roberta.large model (Liu et al., 2019)[4]. According to the Roberta instruction, the training and validation samples were binarized to the desired format. We fine-tuned only weight-decay and dropout coefficients due to heavy performance costs. The learning rate was set equal to 1e-05. All models were trained for 20 epochs. Validation occurred at the end of each epoch. Roberta models were trained at the sentence level without using all sentences from the window.

3.3 Subtask 2. Named entity recognition

The second subtask was named entity recognition in the definition domain. Entity labels were selected among various definitions and term types. Entities could span across several words. In the experiment, the model was trained at the window level.

We relied on the code by Kamal Raj [5]. The BERT-large-uncased model was used. For each token in the example, we took BERT embedding from the first subtoken and passed it through a dropout layer followed by a linear layer. Cross entropy was used as an error function. All non-entity tokens were ignored for loss function calculation. The labels '[CLS]' and '[SEP]' were used to mark the beginning and the end of each example. We also optimized the learning rate, dropout rate, and weight decay coefficients using the validation dataset.

4 Results

Model	w1	w2	w3	Task1 Dev F1	Task1 Test F1	Task2 Dev F1	Task2 Test F1	Task3 Dev F1	Task3 Test F1
m.-task BERT ♠	1.0	1.0	1.0	0.813	**0.848**	-	-	0.631	0.604
m.-task BERT ♠	1.0	0.1	1.0	0.818	0.830	-	-	**0.667**	**0.723**
m.-task XLNet ♠	0.0	0.0	1.0	-	-	-	-	0.456	0.655
m.-task XLNet ♠	1.0	0.0	1.0	**0.839**	0.826	-	-	0.463	0.48
m.-task BERT ♣	0.0	1.0	-	-	-	0,652	0.541		
m.-task BERT ♣	1.0	0.3	-	0.838	0.807			-	-
s.-task BERT ♢	-	1.0	-	-	-	**0.656**	**0.581**	-	-
s.-task Roberta	1.0	-	-	0.805	0.816	-	-		

Table 1: The best results achieved by our models on the test set. ♠ denotes a model trained jointly on all three subtasks which require the knowledge of entity spans, ♣ denotes a model trained on first and second subtasks and ♢ denotes a model trained only on the first subtask. Roberta model opposite to all other models was trained only on single sentences on the first subtask.

We ranked 5th in the task of sentence classification and 23rd and 21st in named entity recognition and relation classification. Our system achieved 0.844 in the F1 metric for the first task with the difference from the first place equal to approximately 0.03 points. For the second task our final result score was equal to 0.52 while the difference amounted to 0.32. For the third task our F1-score was equal to 0.61.

In Table 1 we provide the post-evaluation results of our models for all three subtasks. Entity spans for subtask 1 models (denoted by ♠) were inferred from predictions of our best single-task model for subtask 2 on the development dataset.

For the first task we tried a Roberta based model and BERT and XLNet-based multi-task learning models. Multi-task approach outperforms the Roberta model for this task. It is true not only for the best model but for all multi-task models trained on all three subtasks where the sentence weight is not set to 0. However, we could not improve single-task results for named entity recognition. It might be due to

[4]https://github.com/pytorch/fairseq/tree/master/examples/roberta
[5]https://github.com/kamalkraj/BERT-NER/

insufficient training time because the task itself is more difficult than binary classification. XLNet and BERT results turned out to be close to each other. Their exact results may depend on a lot of factors such as seed number which are not covered in the article. Multi-task learning results with different weighting schemes can be seen in the Appendix.

4.1 Error analysis (Subtask 1)

Type	Description	Window example	% in wrong answers	% in correct answers
1a	w/o found entities in whole context	`Usually hybrids tend to be less fit ; therefore , such reproduction diminishes over time , nudging the two species to diverge further in a process called reinforcement .`	neg: 0 pos: 14.1	neg: 28.2 pos: 0
1b	w/o found entities in sentence	III 6078 . For example , under the Fifth Amendment a person can be tried in federal court for a felony — a serious crime — only after a grand jury issues an indictment indicating that it is reasonable to try the person for the crime in question . III (Term {A grand jury} is Definition {a group of citizens charged with deciding whether there is enough evidence of a crime to prosecute someone} .)	neg: 0 pos: 28.2	neg: 63.5 pos: 0
2a	negative sentences similar to positive	III 2916 . Another type of microscope Definition {utilizing wave interference and differences in phases to enhance contrast} is called Term {the phase contrast microscope} . III While its principle is the same as the interference microscope , the phase - contrast microscope is simpler to use and construct .	neg: 9.4 pos: 0	neg: 1.2 pos: 0
2b	positive sentences similar to negative	2128 . Although marked by great topographic , linguistic , and cultural diversity , this region cradled a number of civilizations with similar characteristics . Mesoamericans were Term {polytheistic ; Definition {their gods possessed both male and female traits and demanded blood sacrifices of enemies taken in battle or ritual bloodletting} . III Corn , or Alias-Term {maize} , domesticated by 5000 BCE , formed the basis of their diet . III	neg: 0 pos: 14.1	neg: 0 pos: 0
3	contains **which/is/are/that**	6454 . These ideas become part of the citizens ' frame of reference and affect their decisions . III Lippmann 's statements led to Term {the hypodermic theory} , which Definition {argues that information is " shot " into the receiver 's mind and readily accepted} . III Walter Lippmann . 1922 .	neg: 31.8 pos: 25.9	neg: 36.5 pos: 21.2

Figure 4: Types of the examples our model struggles most with. There were in total 85 wrongly predicted examples. 85 examples were additionally randomly selected from the correct predictions. *pos* and *neg* denotes positive and negative classes. For example, 14.1% of wrong answers are of type *1a* from the positive class and 0% are of type *1a* from the negative class

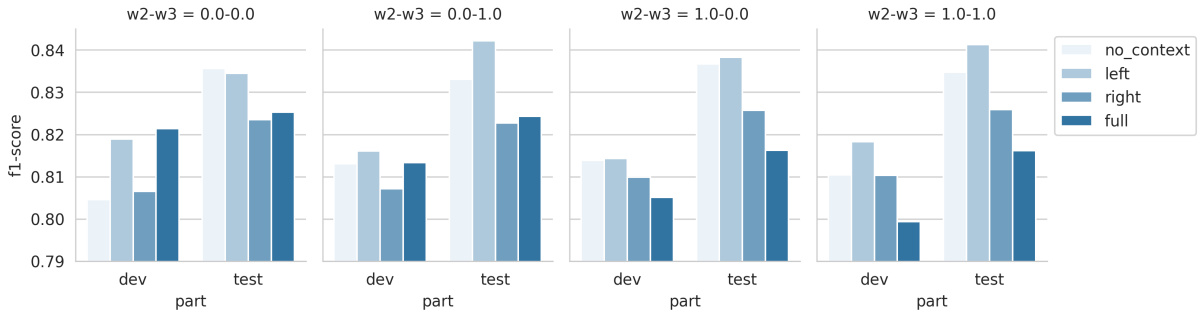

Figure 5: Ablation analysis of contexts. w2-w3 = 1.0-1.0 mean that the weights of the second and the third subtasks are equal to 1.0.

In Figure 4 we show our classification of the main error types of our best-performing relation extraction models for the first subtask. We manually labelled misclassified examples according to their error type. It turns out that most errors come from our model being too sensitive to words typical for definitions, e.g. various conjunctions (which, that). Another major downside is mishandling of named entities. It proves that named entity information might be helpful for telling whether a message contains a definition.

After the shared task we have also studied the influence of context on model results for the first subtask (see Figure 5). The texts in the dataset were split at a sentence level by the organizers. So we decided to see how full text inputs influenced the results. Three context types were studied: no_context, left, right and full. The context shows which information is left with respect to the analyzed sentence. Left context means that we make predictions for the sentence and all words to the left. Full context means that we preserve the sentence and all words to the left and right. We filter examples which have main entity outside of preserved words of the window. We did early stopping by best f1-score for positive class on first subtask. As can be seen from Figure 5, leaving only left context improves model results. The fact that full context is performing poorly relative to other variants maybe attributed to the filtering process for examples with main entity out of context.

5 Conclusion

This work is about our results in DeftEval challenge which was devoted to finding and classifying definitions in texts. A single Transformer-based model was adopted for both tasks simultaneously. We ranked 5th in the task of sentence classification and 23rd and 21st in named entity recognition and relation classification. In this paper we describe our system and analyze the errors.

Acknowledgements

We thank the organisers of the competition for such an inspiring task. We are grateful to our reviewers for their useful suggestions. The contribution of Nikolay Arefyev to the paper was partially done within the framework of the HSE University Basic Research Program funded by the Russian Academic Excellence Project '5-100'.

References

Livio Baldini Soares, Nicholas FitzGerald, Jeffrey Ling, and Tom Kwiatkowski. 2019. Matching the Blanks: Distributional Similarity for Relation Learning. In *arxiv.org*, pages 2895–2905.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. oct.

Yiping Jin, Min-Yen Kan, Jun-Ping Ng, and Xiangnan He. 2013. Mining scientific terms and their definitions: A study of the ACL anthology. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 780–790, Seattle, Washington, USA, October. Association for Computational Linguistics.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S Weld, Luke Zettlemoyer, Omer Levy, and † Allen. 2019. SpanBERT: Improving Pre-training by Representing and Predicting Spans. Technical report.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Roberto Navigli, Paola Velardi, Juana María Ruiz-Martínez, et al. 2010. An annotated dataset for extracting definitions and hypernyms from the web. In *LREC*.

Sasha Spala, Nicholas A. Miller, Yiming Yang, Franck Dernoncourt, and Carl Dockhorn. 2019. DEFT: A corpus for definition extraction in free- and semi-structured text. In *Proceedings of the 13th Linguistic Annotation Workshop*, pages 124–131, Florence, Italy, August. Association for Computational Linguistics.

Sasha Spala, Nicholas Miller, Franck Dernoncourt, and Carl Dockhorn. 2020. Semeval-2020 task 6: Definition extraction from free text with the deft corpus. In *Proceedings of the 14th International Workshop on Semantic Evaluation*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Adv. Neural Inf. Process. Syst.*, volume 2017-Decem, pages 5999–6009.

Amir Pouran Ben Veyseh, Franck Dernoncourt, Dejing Dou, and Thien Huu Nguyen. 2020. A joint model for definition extraction with syntactic connection and semantic consistency. In *AAAI*, pages 9098–9105.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in neural information processing systems*, pages 5754–5764.

A Multi-task learning on subtasks 1, 2

Figure 6 shows the results of the models trained jointly on the first and the second subtasks. We set the weight of the target subtask to 1.0, while changing the weight of another subtask. For subtask 1 multi-task learning seems beneficial on the dev set, which was used for early stopping. From the test set performance we see that this improvement is comparable with the variance of the scores. For subtask 2 multi-task learning evidently hurts.

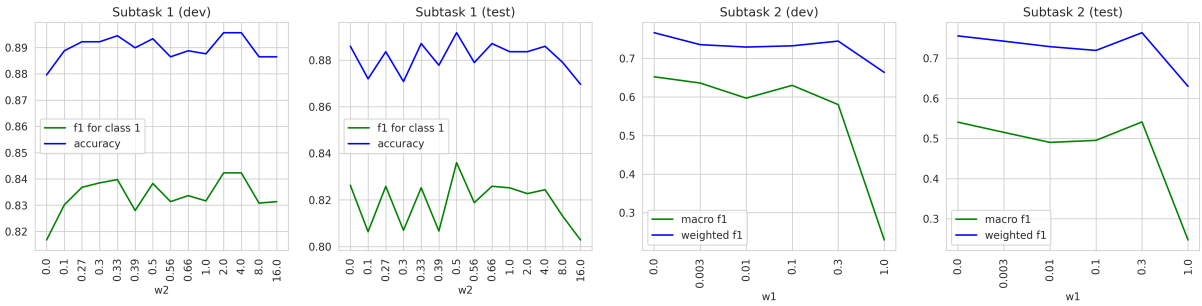

Figure 6: Scores for subtask 1 and subtask 2 with regard to each others weights.

B Multi-task learning on all three subtasks

In Figure 7 you can see the results of the models trained jointly on all three subtasks. For the first subtask we used entity spans predicted by our best single-task sequence labeling model, while for the third subtask we used gold entity spans. Also, for subtask 1 we did early stopping by the F1 score of the positive on the subtask 1 development set.

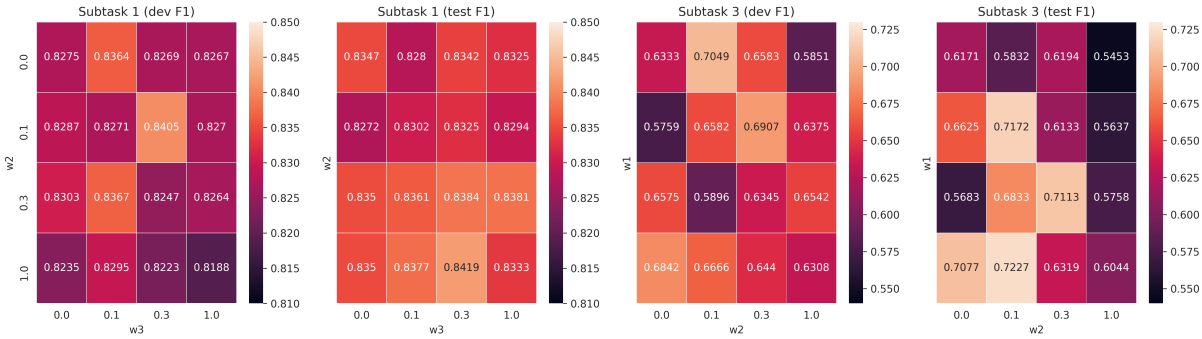

Figure 7: Scores for subtasks 1, 3 with regard to the weights of the other two subtasks.

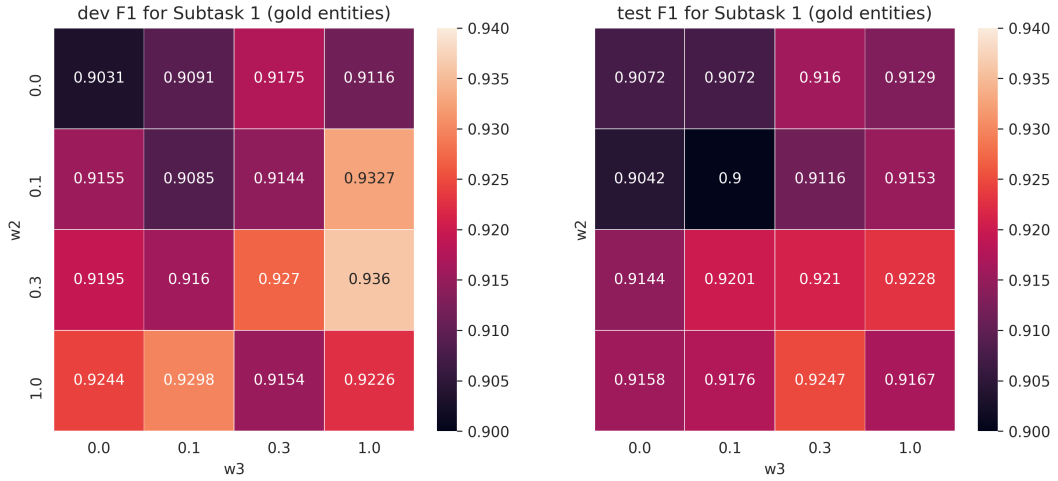

Figure 8: F1 scores for subtask 1 with regard to the weights of the other two subtasks. Gold entity spans are fed to the model.

Figure 8 shows the results for subtask 1 when the gold entity spans are used instead of the predicted ones. It seems, that our results for subtask 1 could be way better if our model for subtask 2 was better at predicting entities spans.