ECNUICA at SemEval-2021 Task 11: Rule based Information Extraction Pipeline

Jiaju Lin, Jing Ling [*], Zhiwei Wang[*], Jiawei Liu[*], Qin Chen, Liang He
School of Computer Science and Technology
East China Normal University
51205901095@stu.ecnu.edu.cn

Abstract

This paper presents our endeavor for solving task11, NLPContributionGraph, of SemEval-2021. The purpose of the task is to extract triples from a paper in the Nature Language Processing field for constructing an Open Research Knowledge Graph. The task includes three sub-tasks: detecting the contribution sentences in papers, identifying scientific terms and predicate phrases from the contribution sentences; and inferring triples in the form of (subject, predicate, object) as statements for Knowledge Graph building. In this paper, we apply an ensemble of various fine-tuned pre-trained language models (PLM) for tasks one and two. In addition, the self-training methods are adopted for tackling the shortage of annotated data. For the third task, rather than using classic neural open information extraction (OIE) architectures, we generate potential triples via manually designed rules and develop a binary classifier to differentiate positive ones from others. The quantitative results show that we obtain the 4^{th}, 2^{nd}, and 2^{nd} rank in three evaluation phases.

1 Introduction

The notion of Open Research Knowledge Graph (ORKG) is first proposed by (Jaradeh et al., 2019) who take steps toward a knowledge graph based infrastructure that acquires scholarly knowledge in machine actionable form. In that form, researchers can keep up with cutting edge academic achievements and eliminate cognitive overload. To accelerate the construction of ORKG, an automatic system is expected. The SemEval-21 task11 is a triple extraction task targeted at building that system. As shown in Table1, the task is divided into three sub-parts corresponding to different processing steps: Sub-task A detects contribution sentences in English articles and classifies them into information units such as $\mathrm{Approaches}$, Models, and $\mathrm{Ablation-analysis}$; Sub-task B extracts scientific terms and relational cue phrases from contribution sentences; Sub-task C infers subject-predicate-object triples for KG building with the results of two previous sub-tasks.

For Sub-task A—contribution sentence detection—the evaluation data covers a larger sphere than the training data. Additionally, the amount of annotated samples differs among research fields. Hence, we use self-training to generate a set of silver samples for fields lacking gold data. An ensemble of fine-tuned PLM based classifiers is then deployed to categorize sentences. For sub-task B—scientific term extraction—we compared BERT based sequence labeling systems in detail and chose the best architecture. For sub-task C—triple generation—we give insight into the construction of triples and designed a rule for potential triples generation. A binary classifier is then applied to distinguish the positive triples.

Our quantitative results show data augmentation via self-training is of paramount importance for sub-task A. Although seldom is CRF used with transformer-based language models together, in the system for sub-task B, an additional CRF layer after a RoBERTa based encoder can still boost performance. In sub-task C, popular neural information extraction models are inferior to the rule based methods.

2 Background

2.1 Data description

The training process is developed on the dataset provided by the SemEval21 Task11. The training dataset involves 237 papers from 24 fields of natural language processing, organized hierarchically with contribution sentences, info units, entities, and

[*]equal contribution

Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021), pages 1295–1302
Bangkok, Thailand (online), August 5–6, 2021.

Objects to Identify	Exampls
Sentence	We use the $\mathrm{BERT}_{\mathrm{BASE}}$ model pre-trained on English Wikipedia and BooksCorpus for 1M steps.
Information Unit	model
Scientific Term and Predicate Phrases	used,$\mathrm{BERT}_{\mathrm{BASE}}$ model, pre-trained on,English Wikipedia, BooksCorpus,for,1M steps
Triples	(Contribution, has, ExperimentalSetup), (ExperimentalSetup, used, BERTBASE model), ($\mathrm{BERT}_{\mathrm{BASE}}$ model, pre-trained on, English Wikipedia), ($\mathrm{BERT}_{\mathrm{BASE}}$ model, pre-trained on, BooksCorpus), ($\mathrm{BERT}_{\mathrm{BASE}}$ model, for, 1M steps)

Table 1: Objects need to be identified

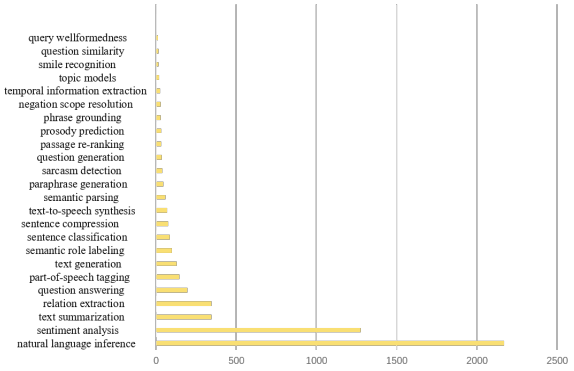

Figure 1: Numbers of annotated sentences in each domain

triples. Thus, for different tasks we can use disparate parts of the dataset.

Inherent challenges also come with data collection. As shown in Fig 1, first, there is a dramatic discrepancy among the number of annotated papers. The NaturalLanguageInference field received the richest resources. A total of over one hundred papers in this area are annotated. On the other hand, for the domains with poor annotation like PhraseGrounding and QueryWellformedness, only a single paper is provided. Moreover, a postdoctoral researcher with a background in natural language processing is responsible for finishing the pilot annotation task (D'Souza and Auer, 2020). Therefore, the annotated data is relatively subjective and sometimes even inconsistent. For example, some information units nested in Experiments actually also included a combination of ExperimentalSetup and Results. Alternatively, it can be combination of Tasks and their Results.

2.2 Related Work

A vast amount of excellent work has been done in the areas of these subtasks. Early work employing CNNs, RNNs and attention based RNN or CNN models has made great progress in sentence classification tasks. (Yang et al., 2017; Liu and Zhang, 2017). Tai et al. (2015) inspired innovation in traditional LSTM networks. The tree-LSTM structure mentioned in their paper is enhanced with dependency or constituency trees. Since Graph Neural Network (GNN) is first used for sentence classification tasks, GNN has been one of the most prevalent encoders for Natural Language Processing(NLP) tasks. Transformer based models are also popular encoders. They are so powerful that they have even been widely used in computational vision areas (Dosovitskiy et al., 2020).

The sequence labeling task is a critical component of NLP applications. There are two basic approaches. In the token level approach, a sequence of tokens is used as an input of sequence tagging models, and tags for each token can be output. Other approaches attempt to solve problem on the sentence level. Lu and Roth (2015) designed a hypergraph, which provides a resolution for the discontinuous terms.

Traditional open information extractors are based on rules and statistical approaches, like Stanford-IE(Angeli et al., 2015),OpenIE-5(Saha and Mausam, 2018) and MinIE (Gashteovski et al., 2017).These methods apply semantic parsers combined with predefined rules to extract triples. Recently, neural OpenIE methods dominate this research field. RnnOIE (Stanovsky et al., 2018) inspired by the sequence labeling systems identifies relation phrases first then combine relations with

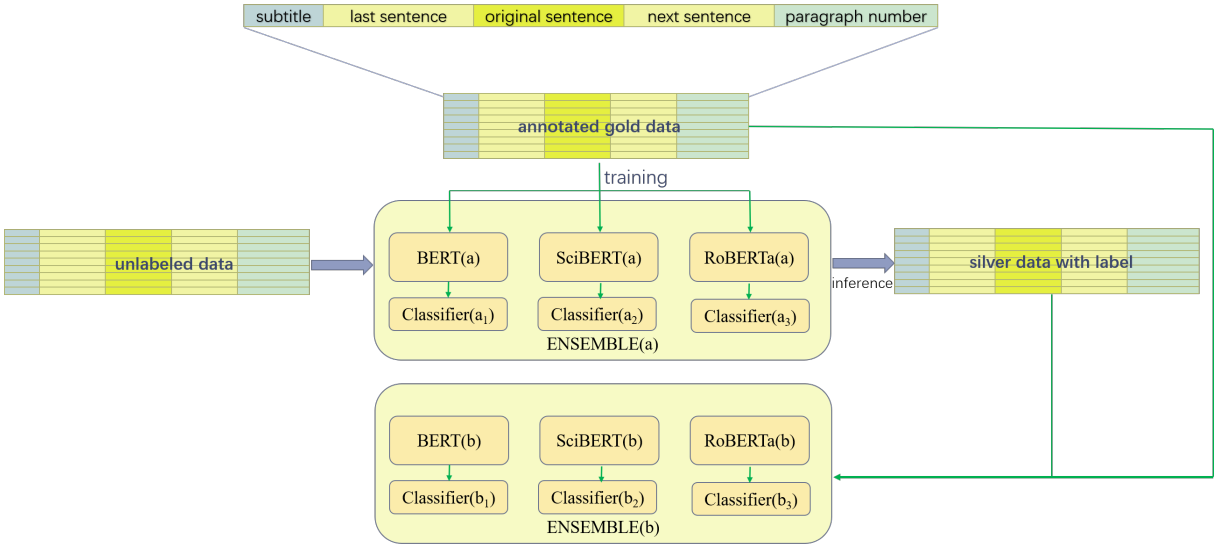

Figure 2: An overview of the system for sentence classification

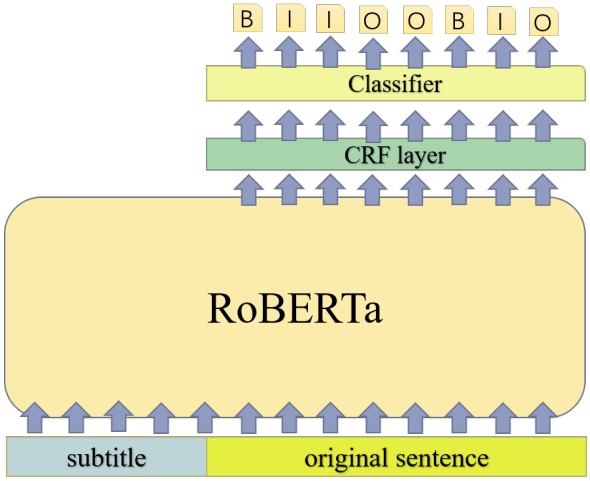

Figure 3: An overview of the system for scientific term extraction

arguments. IMOJIE (Kolluru et al., 2020) takes advantage of seq2seq architectures. It is trained on training data bootstrapped from extractions of several tradition systems such as Stanford-IE.

3 System Description

Systems applied for contribution sentence detection and scientific term extraction are based on the RoBERTa (Liu et al., 2019),SciBERT (Beltagy et al., 2019) and basic BERT model with task-specific modifications. For triple classification tasks, a SciBERT(Beltagy et al., 2019) based model is used for candidate triple classification. Moreover, self-training, ensemble and rule design enhanced system performance in different ways.

3.1 Contribution Sentence Detection

The contribution sentence detection task is handled as a sentence classification (SC)problem. Let U be the union of a predefined sentence type set and ϵ indicate that the sentence is not a contribution sentence. According to the task description provided by D'Souza et al. (2021), one contribution sentence could belong to one of eleven categories, called info units. Hence U has twelve elements with ϵ added. As shown in Fig2, the input data consisted of four parts: the original sentence, contextual information, a sub-title of the paragraph and the number of paragraph, with the separator token ([SEP]) in between. For contextual information, we used the adjacent sentences of the original one. We define the sub-title of a paragraph as the nearest title found previous to the begin of this paragraph. Besides, the paragraphs are numbered from zero following an increasing order. We add [CLS] token at the top of the sequence and build a classifier on top of its embedding, which is generated from BERT based model, similar to what Devlin et al. (2019) did for pre-training.

Inspired by incremental semi-supervised training(Rosenberg et al., 2005), we introduced the similar training process. Prior to that, we need to prepare the additional unlabeled data. Newest papers are downloaded according to the areas then transformed into Stanza version as the papers provided in training data. The amount of additional data for every field is about ten articles. Training process takes the following steps: first, train BERT based models that will be used in ensemble on gold

sentence	triples
We also apply 3 dense blocks based on char - ResNet which we refer to as char - DenseNet, to compare the difference between residual connection and dense connection.	(Baselines, apply, 3 dense blocks) (3 dense blocks, based on, char - ResNet) (char - ResNet, refer to as, char - DenseNet)

Table 2: A sentence and triples from it

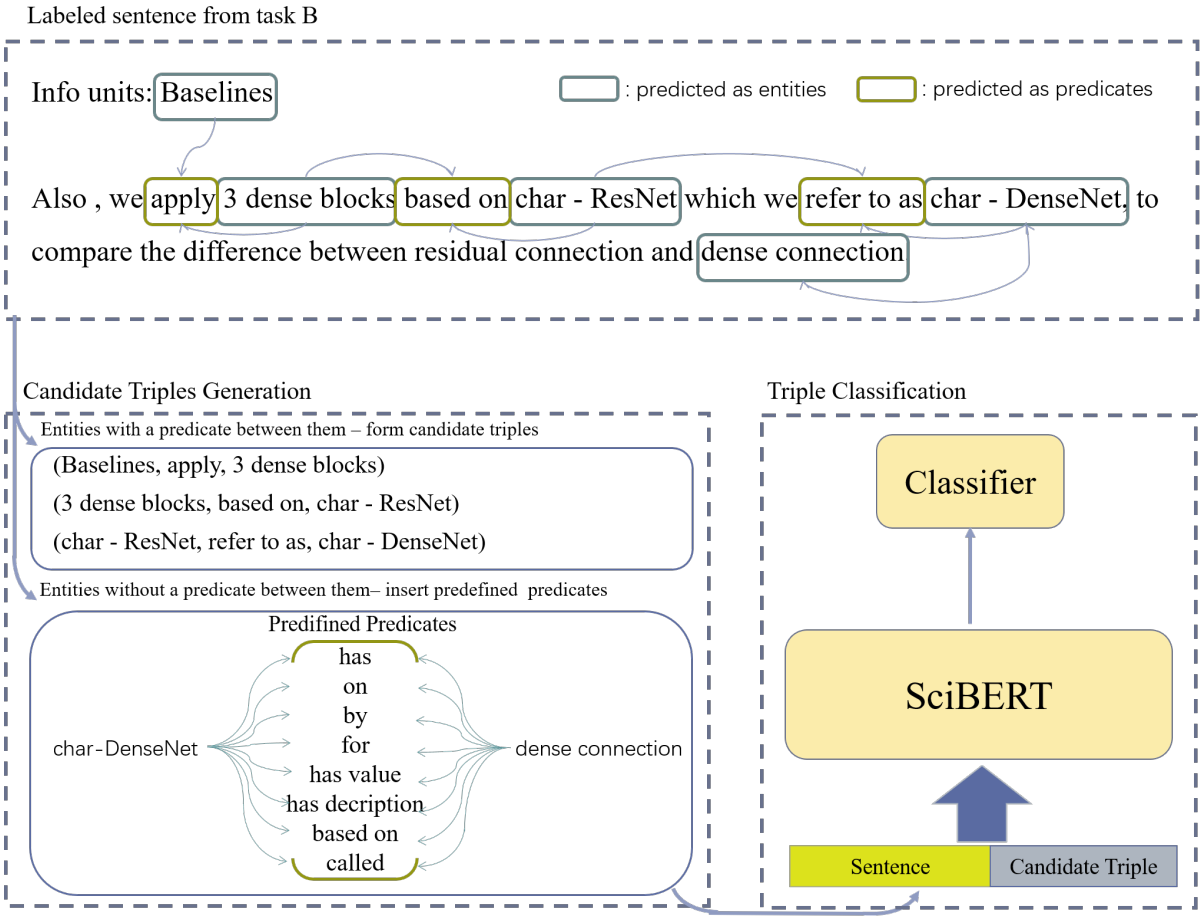

Figure 4: Model overview for triple generation

data, that is the annotated data in dataset, until parameters converged. Next, put these models in an ensemble to tag the unlabelled data for areas suffered from data insufficiency. In the rest part of this passage, we call data labeled by our models 'silver samples' or 'silver data'. Only the sentences all models in the ensemble labelled unanimously would be deemed as silver samples and used for further training. After that, a combination of gold and silver data is input for training another three models with unchanged hyperparameters. Above progress iterates until no further progress could be achieved. That means the iteration is stopped when loss becomes stable. Thus, three fine-tuned BERT based models are ready for inferring and we collect them in an ensemble. When inferring, with a sentence input, each model outputs a vector V, of twelve dimensions. We calculate a weighted sum of three vectors and use the index of the max element to determine the final class of a sentence. Weights here are hyperparemeters and chosen manually.

$$V = 0.25*V_{SciBERT}+0.2*V_{RoBERTa}+0.1*V_{BERT}$$

3.2 Scientific Term Extraction

We consider scientific term extraction as a sequence labeling(SL) task. Specifically,as Fig3 shows, a RoBERTa exploiting CRF layer marks every subword token of the input sentence with one of the label in B,I,O, where 'B' is the beginning of a term, 'I' indicates that the token is inside the term and

'O' indicates that the token stands outside of the term. In this way,one sequence of a 'B' followed by a continuing sequence of 'I' is recognized as a legal term. Input sequences are also engineered. The sub-title information is attached to the raw sentence. To facilitate triple generation, two models are trained for labeling predicate and entity individually. Discrimination of types of terms allows the system to exploit the information of a predicate more efficiently.

3.3 Triple Generation

Given the fact that the outcomes of supervised open information extraction architectures are not as expected, we make use of the result of term extraction. As shown in Table 2, triples from a sentence always overlap at the head and tail. Given that fact, when generating a candidate triple, a predicate acts as the anchor and its neighboring entities are seen as the subject and object, following the sentence reading order, as demonstrated in Fig4. It is rare that the subject appeared later than the object in a sentence.

At times, as task description paper (D'Souza et al., 2021) mentioned, when no suitable predicate phrases could be inferred from the sentence, one candidate from a pre-defined set of predicates could be utilized. The set including "has", "on", "by", "for", "has value", "has description", "based on", "called". We call these triples with predefined predicates "special triples". A greedy matching is introduced that each predefined predicate is inserted between every adjacent entity pair to compose a potential special triple. To illustrate, take the entity pair "char-DenseNet" and "dense connection" as an instance. In Fig4, as the result of term extraction shows that there is no phrase labelled as predicate between "char-DenseNet" and "dense connection". To form candidate triples for this entity pair , each predicate in the predefined predicate set is inserted between the entity pair . After all potential triples are generated, gather unions of each candidate triple and the sentence where the triple came from as input data. A SciBERT based binary classifier then judge if a union is rational. We refer to it as the 'candidate triple judge model' in the rest of this article.

Additionally, another rule is designed for cross-sentence triples, which takes up three percent of all triples. Such amount cannot be ignored also. We observe that when only one term can be extracted from a sentence, it is highly possible that the term is a composition of a cross-sentence triple. Such terms adjacent to each other are integrated into a cross-sentence triple according to the subject-predicate-object order. From example, if there are three adjacent contribution sentences that we can only extract one phrase from each, and these three phrases are predicted as entity,predicate and entity respectively. Thus we can combine them together as a cross-sentence triple. For these triples, we do not apply a further filter and add them into the final output directly.

Hparam	**SC**	**TE**	**TG**
Number of epochs	8	20	10
Max length	200	128	256
Batch size	32	16	32
Learning rate	2e-5	1e-5	1e-5
Optimizer	AdamW		
Loss	cross entropy		

Table 3: Hyperparameters for models. SC means sentence classification, TE stands for term extraction and TG is the abbreviation of triple generation

4 Experimental Setup

In this section, we describe the models we used in the final submission and their parameters in detail. It should be possible to reproduce our work.

4.1 Models and Parameters

Before training, we divide papers into three groups: train set, dev set, and test set, with a ratio of 8:1:1. We then mix all sentences in each set together. In this way, data leakage is prevented. Otherwise, sentences in test set and train set could come from the same article. The hyperparameters for training are shown in Table3. Our implementation uses only Pytorch for the first two sub-tasks' models and AllenNLP for the last candidate triple judge model.

For Contribution Sentence Classification task, we attempt to take advantage of diverse models. An ensemble of BERT, SciBERT and RoBERTa is applied. During the training process, F1 score on dev set works as a criteria for choosing the best epoch and model weights. Additionally, because our model is consistently confused between the info units of Approach and Model, we convert sentences with the word 'approach' to the unit Approach after receiving predictions from a neural network. For Scientific Term Extraction, we used

	Contribution Sentence Detection	
	sentence detection	information unit classification
F1 of ensemble	0.3978	0.8108
F1 of SciBERT	0.3856	0.8049

	Scientific Term Extraction		Triple Generation	
	RoBERTa+CRF+BIO	RoBERTa+span tagging	rule based method	IMOJIE
F1	0.7774	0.7567	0.4473	0.1729

Table 4: F1 scores of models. For the submitted model, the F1 scores are from the leader board , for the baseline model the F1 scores are from results on dev set

RoBERTa as the encoder and elaborated more on different decoders. The basic BIO tagging model performs better than the span based one. When training the potential triple judge model, the learning rate rises first then falls following the method used by Vaswani et al. (2017)

4.2 Baselines

We endeavor to search for the best baselines. For sentence classification, we use single SciBERT model as our baseline, while for sequence tagging, we employ RoBERTa without CRF layer and rather using span tagging decoder as a strong baseline. In the triple generation task, we once tried to employ neural open information extraction models, so IMOJIE can be deemed as a baseline.

4.3 Evaluation Metric

We use F1 value as the main metric, the average F1 is arithmetic mean value of sentence F1, terms F1, info units F1, and triples F1. When computing triples F1, strict standard is employed. Only when every division of a predicted triple matched the gold answer, it can be counted as a correct inference.

$$F1 = \frac{2 * P * R}{P + R} \tag{1}$$

where P means precision of the prediction and R means recall.

$$\begin{aligned} F1_{avg} =& avg(F1_{sentence}, F1_{terms}, \\ & F1_{infounits}, F1_{triples}) \end{aligned} \tag{2}$$

5 Results

With the gold data of the upstream task provided, the F1 value of sentence, info units,terms, and triples are 0.3978, 0.8108, 0.7774, and 0.4473 respectively, as shown in Table4 .

The enhancement in the sentence classification is clear. As a prevalent technology, ensemble has become a necessary part of algorithm competitions. The only restriction is that all models in ensemble should have an F1 over fifty percent. With the help of data augmentation, the info unit classification result occupied 2^{nd} position in the final ranking.

It marveles us that the model with span based tagger performed worse than the BIO tagger. Many NER experiments shows the evidence that the span based tagging decoder outperforms the simple BIO tagger. We believe the main reason is that, when exposed to the data in this task, there is no need to discern types of entities. While the span based decoder are equipped with the ability to infer entity types, such design may be suboptimal and create additional errors.

To some extent, the improvements in the triple generation task proves that neural OIE models are inapplicable to the task on this dataset. The main cause may be the different definitions of 'predicate.' In our task, prepositions always appear in the position of predicate in triples. Likewise, the subject and object are persons or specific terms while for sentences in science papers only scientific notions can be found. Given so much elaboration in our system, terms extraction and triple generation task also achieved the second place on the leaderboard.

6 Conclusion

We engaged in SemEval-2021 task11 NLPContributionGraph with models integrating features suitable for disparate tasks. We took insight on the impact of different parts on the final results, fine-tuned hyperparameters, and attempted various feature engineering methods. We ranked 4^{th}, 2^{nd}, and 2^{nd} in three evaluation phases and our final model demonstrated its superiority over several strong baselines.

Acknowledgments

The task is sponsored by ICA group, East China Normal University. Besides, thanks are due to Yi-

wei Yan for her supervision as without her help it would have been impossible to finish writing so quickly.

References

Gabor Angeli, M. Johnson, and Christopher D. Manning. 2015. Leveraging linguistic structure for open domain information extraction. In *ACL*.

Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. SciBERT: A pretrained language model for scientific text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3615–3620, Hong Kong, China. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. 2020. An image is worth 16x16 words: Transformers for image recognition at scale.

Jennifer D'Souza, Sören Auer, and Ted Pedersen. 2021. SemEval-2021 task 11: Nlpcontributiongraph - structuring scholarly nlp contributions for a research knowledge graph. In *Proceedings of the Fifteenth Workshop on Semantic Evaluation*, Bangkok (online). Association for Computational Linguistics.

Jennifer D'Souza and Sören Auer. 2020. Nlpcontributions: An annotation scheme for machine reading of scholarly contributions in natural language processing literature.

Kiril Gashteovski, Rainer Gemulla, and Luciano del Corro. 2017. MinIE: Minimizing facts in open information extraction. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2630–2640, Copenhagen, Denmark. Association for Computational Linguistics.

Mohamad Yaser Jaradeh, Allard Oelen, Kheir Eddine Farfar, Manuel Prinz, Jennifer D'Souza, Gábor Kismihók, Markus Stocker, and Sören Auer. 2019. *Open Research Knowledge Graph: Next Generation Infrastructure for Semantic Scholarly Knowledge*, page 243–246.

Keshav Kolluru, Samarth Aggarwal, Vipul Rathore, Mausam, and Soumen Chakrabarti. 2020. IMoJIE: Iterative memory-based joint open information extraction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5871–5886, Online. Association for Computational Linguistics.

Jiangming Liu and Yue Zhang. 2017. Attention modeling for targeted sentiment. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 572–577, Valencia, Spain. Association for Computational Linguistics.

Y. Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, M. Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *ArXiv*, abs/1907.11692.

Wei Lu and Dan Roth. 2015. Joint mention extraction and classification with mention hypergraphs. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 857–867, Lisbon, Portugal. Association for Computational Linguistics.

C. Rosenberg, M. Hebert, and H. Schneiderman. 2005. Semi-supervised self-training of object detection models. In *2005 Seventh IEEE Workshops on Applications of Computer Vision (WACV/MOTION'05) - Volume 1*, volume 1, pages 29–36.

Swarnadeep Saha and Mausam. 2018. Open information extraction from conjunctive sentences. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2288–2299, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Gabriel Stanovsky, Julian Michael, Luke Zettlemoyer, and Ido Dagan. 2018. Supervised open information extraction. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 885–895, New Orleans, Louisiana. Association for Computational Linguistics.

Kai Sheng Tai, Richard Socher, and Christopher D. Manning. 2015. Improved semantic representations from tree-structured long short-term memory networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1556–1566, Beijing, China. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need.

Min Yang, Wenting Tu, Jingxuan Wang, Fei Xu, and Xiaojun Chen. 2017. Attention based lstm for target dependent sentiment classification.