Focused Contrastive Training for Test-based Constituency Analysis

Benjamin Roth, Erion Çano
Digital Philology
Research Group Data Mining and Machine Learning
University of Vienna, Austria
first.last@univie.ac.at

Abstract

We propose a scheme for self-training of grammaticality models for constituency analysis based on linguistic tests. A pre-trained language model is fine-tuned by contrastive estimation of grammatical sentences from a corpus, and ungrammatical sentences that were perturbed by a *syntactic test*, a transformation that is motivated by constituency theory. We show that consistent gains can be achieved if only certain positive instances are chosen for training, depending on whether they could be the result of a test transformation. This way, the positives, and negatives exhibit similar characteristics, which makes the objective more challenging for the language model, and also allows for additional markup that indicates the position of the test application within the sentence.

1 Introduction

One way linguists analyze language is by applying *linguistic tests*: Here, transformations that are driven by a specific theory are applied to utterances (e.g., sentences) and if the result of the transformation is judged grammatical (according to the linguist's introspection) then the test has identified the occurrence of a specific category or phenomenon. A prime example of this process is that of *constituency tests*. Constituency theory posits that language is structured hierarchically into constituents, i.e., spans of specific types that function as units and can be moved or replaced by other units of the same type (while other spans that are not constituents do not exhibit those properties). The advantage of test-based linguistic analysis is that it operates very close to the underlying theory, and the principles captured by the tests, stemming from theory, can be very general. In contrast to grammar-based approaches, in a test-based setting, one does not need to spell out all different phenomena that would follow from the theory.

If automated test-based analysis could be successfully implemented, this would theoretically render the laborious manual construction of grammars or training corpora obsolete. However, the test-based linguistic analysis would instead require the correct specifications of tests (transformations) and a grammaticality model (that can replace the linguists' introspection). In this work, we show how to successfully automate a simple class of constituency tests based on *pro-form replacements*. Moreover, we show how a grammaticality model based on annotated training data can be replaced by one using only unsupervised data. We do this using a proposed mechanism that we refer to as *focused contrastive training*.

Our study reveals the following insights: (i) We compare different strategies for combining the scores of different constituency tests and find that taking the maximum works better than the average or voting. (ii) We systematically compare the performance of different pro-forms, and find a selection of 4 pro-forms (out of 18) that perform markedly better than a previously suggested set. (iii) We

35th Conference on Neural Information Processing Systems (NeurIPS 2021), Sydney, Australia.

show that focused contrastive training outperforms a supervised grammaticality model as well as a previously proposed negative sampling scheme.

2 Related Work

There is a growing interest in leveraging PLMs (Pretrained Language Models) like BERT and XLNet (Devlin et al., 2019; Yang et al., 2019) which are based on Transformer self-attention layers (Vaswani et al., 2017). They offer contextual word representations and have proven highly effective in various NLP downstream tasks. Moreover, both PLMs and grammar induction models are trained on the same objective: learn to model natural languages which is useful for several applications. These facts have motivated work to explore the syntactic knowledge captured in PLMs for the possibility of using them in unsupervised constituency parsing: e.g., Mareček and Rosa analyze the encoders of Transformer architectures and design a method for extracting constituency trees from self-attention heads. Comparing those trees with standard syntactic parse trees, they conclude that Transformers do indeed capture syntax. Similarly, Li et al. (2020a) also extract constituency trees from PLM attention heads. They later rank them based on their properties, revealing multiple insights which could be useful for future research.

Constituent parsing with PLMs has been proposed in both supervised (Kitaev and Klein, 2018; Zhang et al., 2020) and unsupervised settings (Kim and Lee, 2020; Li et al., 2020b). Unsupervised constituency parsing is challenging but offers the possibility to work with an unlimited amount of text, overcoming the need for human supervision. Kim and Lee (2020) utilize unsupervised parsing via PLMs by introducing zero-shot consistuency parsing with a chart-based method. It considers every phrase subspan to judge the plausibility of that phrase. They propose an ensemble technique that selects the top-K PLM attention heads and boosts performance. Cao et al. (2020) propose a semi-supervised approach where they automate a range of movement- and replacement-based constituency tests. Their model is initialized with supervised (Warstadt et al., 2019) and unsupervised grammaticality models and fine-tuned to optimize the likelihood of an unsupervised chart parser. In contrast, in our work, we study self-supervised training of grammaticality models that can be used as an isolated modular component in test-based linguistic analysis (independently of parsing algorithms).

3 Approach

3.1 Constituency analysis with linguistic tests

A general approach to linguistic analysis is to reformulate and replace parts in question with prototypical realizations of a phenomenon, and then judge the result with respect to its grammatical acceptability. If such reformulations are formalized according to a linguistic theory, standardized and operationalized, they can be called a *linguistic test*.

Given a set of transformations (tests) T each test $t \in T$ is a function that takes a word sequence (utterance or sentence) s as well as a contained subspan x as an input and outputs a transformed word sequence $s^{tx} = t(s, x)$. Moreover, there is a real-valued function α where $\alpha(s') > \alpha(s'')$ for two sentences s' and s'' iff s' is more acceptable than s''.

A constituent is any group of words that function as a single unit in a hierarchical structure. One main type of constituent tests are *pro-form substitution tests:* Can the constituent candidate be replaced by a pro-form of the same category? The testing procedure is therefore to (1.) substitute the constituent candidate (2.) judge whether the sentence is still acceptable. In the following, we study how different choices for automating those two steps influence the quality of automated test-based constituent analysis.

3.2 Pro-forms

We created an initial list of pro-forms to cover the most common syntactic categories and include the most common pronouns, including some variability in order to allow for agreement in different contexts. This list also includes the pro-forms suggested by Cao et al. (2020) (underlined). **Pronouns**: it, ones, this, that, they, I, we, you; **Pro-PPs** (preposition+pronoun): of it, for it, in it; **Pro-VPs**: did so, do that, does that; **Pro-sentences**: it is, that it is; **Pro-adverbs**: there, this way.

3.3 Supervised training

A supervised approach for arriving at a grammaticality model α is to train on labeled grammatical and ungrammatical sentences. Similar to Cao et al. (2020), we train a supervised grammaticality classifier using the CoLA dataset (Warstadt et al., 2019).

3.4 Combining scores of different pro-forms

For a grammaticality model α, and input sentences s, a set of tests T, and a selected subspan x, we compare different strategies for combining the scores. **Maximum**, the highest scoring transformation is used: score$(s, x) = \max_{t \in T} \alpha(t(s, x))$; **Average**, the average grammaticality score of all tests is used: score$(s, x) = \frac{1}{|T|} \sum_{t \in T} \alpha(t(s, x))$; **Voting**, two subspans are not directly compared by a score, but by the number tests that score higher for one of them: $|\{t \in T : \alpha(t(s, x)) > \alpha(t(s, x^{'}))\}| > \frac{|T|}{2}$.

3.5 Contrastive training

In contrastive training with negative sampling (Ebisu and Ichise, 2020; Riedel et al., 2013; Ma and Collins, 2018; Ding et al., 2018; Mikolov et al., 2013), positive training instances are taken from a set of observations, i.e. drawn from the data distribution, while negative training instances are constructed such that they are unlikely to be generated from the data distribution.

Contrastive training has been used to pre-train a grammaticality model (Warstadt et al., 2019), using permutations of sentences as corrupted negative samples, based on the assumptions that observed sentences are grammatical and corrupted sentences are ungrammatical. This insight was incorporated by Cao et al. (2020), who used a simple form of contrastive *pre*-training for a grammaticality model for constituent tests. More precisely, they use corruptions based on applications of constituency tests (applied to randomly selected subspans of observed sentences) as negative samples. Since only a small minority of subspans are constituents, it is likely that the results of these transformations are indeed ungrammatical.[1]

More formally, if C is corpus of observations, then:

- $X_{pos} \sim C$ are the positive instances, a subset of instances uniformly sampled from C without replacement ($X_{pos} = C$ if the entire corpus is used).
- $X^t_{neg} \sim \{t(s, x) : s \in C, x \in \text{subspans}(s)\}$ are instances corrupted by a test transformation $t \in T$. In order to achieve an equal ratio between positive and negative samples, we require that $|X_t| = \frac{|X_{pos}|}{|T|}$ for all $t \in T$. The negative instances are then $X_{neg} = \cup_{t \in T} X^t_{neg}$.

We refer to this scheme as *non-focused contrastive training*, and we take all sentences in the training corpus as positive examples X_{pos}. We create an equal amount of negative samples, by first uniformly sampling a test (a pro-form) $t \in T$ and a subspan of $2 - 4$ tokens for each of the (same) sentences, replacing the selected span by the pro-form. The training or test input to the grammaticality model is the sequence of words without any further markup. At test-time, the grammaticality score $\alpha(s^{tx})$ will indicate how much the trained model estimates a sentence (after applying a test transformation) to have characteristics of naturally occurring sentences vs. the perturbed ones.

3.6 Focused contrastive training

While the setting described above makes sense at the first glance, it is a suboptimal contrastive training scheme in the case of syntactic constituency tests: During training, tests are only applied to the negative samples, hence the negative samples exhibit certain patterns more frequently which stem specifically from the transformations. For this reason, the model might learn to generally give a low grammaticality score whenever a test transformation has been applied. However, at test time there are only transformed sentences, and the transformations of constituent spans should result in a high grammaticality score whenever a constituent has been replaced. Moreover, in the above setting, the grammaticality judgment is about the entire sentence, without any indication about the place that has

[1] Assuming binary branching, there are $n\frac{(n-1)}{2}$ subspans but only n constituents of length ≥ 2. Moreover, not every test is a valid replacement for each constituent type.

been affected by the test transformation. However, when applying a test, the question is whether it was the transformation of a specific span that rendered the sentence ungrammatical.

To overcome these weaknesses, we propose *focused contrastive training*: Our method focuses on *relevant* positive instances that *could be the outcome of a test application*. Since these instances exhibit similar patterns pertaining to specific transformations as the corrupted instance, this makes training more challenging but also avoids problematic reliance on artifacts of training data construction. Similarly, balanced sampling of negative instances explained below, adapts aggregate characteristics of the negative set to those of the positive set. Our method focuses on the part of the sentence that is (or, could have been) affected by a test, providing the relevant information to the model during training and testing. We formalize focused contrastive training:

- $X^t_{pos} \sim \{s \in C : \exists s', x \text{ s.t. } s = t(s', x)\}$ are a sample of those instances observed in the corpus that could also have been the result by applying test t to some hypothetical sequence of words x'. $X_{pos} = \cup_{t \in T} X^t_{pos}$ are the positive instances.
- $X^t_{neg} \sim \{t(s, x) : s \in C, x \in \text{subspans}(x)\}$ are instances corrupted by a test transformation $t \in T$, as before. However, since also the positive instances can now be assigned to a specific test, the negative and positive instances can be balanced per test, which avoids test-specific artifacts to be associated with the positive or negative class, by requiring $|X^t_{neg}| = |X^t_{pos}|$. The negative instances are again $X_{neg} = \cup_{t \in T} X^t_{neg}$.

For a more focused representation of each instance in X^t_{pos} and X^t_{neg}, we define $s^{tx} = \text{markup}(t, s, x)$, where $\text{markup}(t, s, x)$ is the transformed sentence $t(s, x)$ together with any additional information about the specific test applied and/or a real or hypothetical sentence as the input to t. It is important to note that this markup could not be added for positive instances in the non-focused setting. In our case of pronominalization tests, we simply mark the start and end positions of the pronouns associated with test t. The grammaticality model is then used to evaluate $\alpha(s^{tx})$.

Specifically, for *focused contrastive training*, we first create each positive set X^t_{pos} by collecting all sentences in the corpus in which the pro-form of t occurs. The occurrence of the pro-form is then marked with start (<S>) and end (<E>) markers to create the input for the grammaticality model, for example:

[*Since the last time he traveled <S> this way <E> several months ago , he has recanted a series of bold forecasts of a recession .*]

Then, for each test t the corresponding amount of negative examples X^t_{neg} is created by randomly selecting sentences from the training corpus, and replacing a subspan of $2 - 4$ tokens with the pro-form of t, indicated with start and end markers, for example:

[*In the past year , both have <S> this way <E> the limits of their businesses .*]

3.7 Data sets

3.7.1 Development and test set

In order to evaluate whether a grammaticality model scores the resulting replacements of constituents higher than that of non-constituents, we create pairs of two subspans x^c, x^n within the same sentence s, where one is a constituent and the other is not.

For every scoring method, we measure its accuracy by the average number of sentences where $\text{score}(s, x^c) > \text{score}(s, x^n)$. We measure the relative score difference within a sentence in order not to disadvantage scoring models for which the grammaticality score might be strongly impacted by parts of the sentence unrelated to the selected subspan.

Following Chen and Manning (2014); Dyer et al. (2015), we use sections 22 and 23 of the WSJ portion of the Penn Treebank[2] as development and test data, respectively. We keep sentences with at least 3 tokens only and remove the shorter ones. For each sentence, we randomly sample a constituent from the treebank annotation (excluding trivial constituents spanning only one token or the entire

[2]This corpus is licensed via the LDC under catalog number LDC99T42.

pronoun set	accuracy
this way	0.7500
this way, did so	0.7763
this way, did so, of it	0.7883
this way, did so, of it, it	0.7996
it, ones, did so	0.6848

Table 1: Accuracy (dev set) for selected pronoun sets, using the CoLA pre-trained model. Greedy selection according to the *maximum* selection strategy (top), and pronoun set of Cao et al. (bottom).

data	scheme	train-full	train-greedy	train-complement
dev	non-focused	0.8188	0.7075	0.7913
test	non-focused	0.8399	0.7189	0.7841
dev	focused	0.8433	0.8242	0.8206
test	focused	0.8560	0.8152	0.8160

Table 2: Effect of focused contrastive training using different pronoun sets during training (the *greedy* pronoun set is used for scoring).

data	scheme	score-full	score-greedy	Cao et al.
dev	non-focused	0.7075	0.8188	0.7243
test	non-focused	0.7142	0.8399	0.7470
dev	focused	0.7907	0.8433	0.7518
test	focused	0.7922	0.8560	0.7602

Table 3: Comparison of different pronouns sets used for scoring (for the model trained on the *full* pronoun set).

sentences), as well as a non-constituent span with the corresponding length. This results in 1,672 sentences for the development set and 2,348 sentences for the test set.

3.7.2 Training sets and RoBERTa model

CoLA. For supervised training, we use the grammaticality annotations of 10,657 sentences from Warstadt et al. (2019).
WSJ. For contrastive training, we use sections 02-21 of the WSJ portion of the Penn Treebank which total in 37,374 sentences. We do not make use of any constituent annotation, as our goal is the unsupervised training of the grammaticality model.
RoBERTa. For all experiments, we fine-tune a pre-trained RoBERTa-base (Liu et al., 2019) model from the huggingface transformer library,[3] following the hyper-parameter choices of Liu et al. (2019); Cao et al. (2020) for the CoLA task. We train for 10 epochs, and choose the model with the best accuracy on the dev data.

Our code is written in Python version 3.8.10. We ran our experiments on a DGX-1 server with Ubuntu 20.04 GNU/Linux using one Nvidia V100 GPU per experiment. The code execution of each experiment took 2h or less.

4 Results and Discussion

In a first experiment, we investigated the performance of different score combination schemes (*maximum*, *average* and *voting*, see Section 3.4). For this, we used the grammaticality model trained with the supervised CoLA dataset and got the following accuracy scores for three strategies on the dev set: *maximum*: 0.7697, *average*: 0.756, *voting*: 0.7075. Hence, the most effective strategy for combining different tests is taking the *maximum* score. This is intuitive because for detecting a constituent it should suffice that one of the tests is appropriate, and it is to be expected that different tests work well in different contexts. The *maximum* strategy is therefore used for all other experiments.

[3] `https://huggingface.co/transformers/`

	sentence	pred.?	const.?
orig.	Mr. Sim figures it will be easier to turn Barry Wright around since he 's now in the driver 's seat .		
repl.	Mr. Sim figures it will be easier to turn Barry Wright around **this way** .	Yes	Yes
repl.	Mr. Sim figures it will be easier to turn Barry **it** 's seat .	No	No
orig.	Other fund managers were similarly sanguine .		
repl.	Other fund managers **did so** .	Yes	Yes
repl.	Other fund **of it** sanguine .	No	No
orig.	On Saturday night , quite a few of the boys in green and gold salted away successes to salve the pain of past and , no doubt , future droughts .		
repl.	On Saturday night , quite a few of the boys in green and gold salted away successes to salve **did so** .	No	Yes
repl.	**This way** green and gold salted away successes to salve the pain of past and , no doubt , future droughts .	Yes	No

Table 4: Example sentences from the dev data. Given is the original sentence, as well as transformed versions from replacing a selected subspan with the pronoun scored highest (by the CoLA model). The last columns indicate whether the replaced span was predicted to be a constituent, and whether it actually was one (according to the treebank).

The results in Table 1 show that strong improvements can be achieved by selecting a subset of pronouns and that the subset of pronouns we arrived at performs more than $10\,\%$ accuracy points better than the one which was originally proposed by Cao et al. (2020). Table 4 shows some example sentences with selected subspans, together with the pronouns that yielded the highest score when substituted.

Tables 2 and 3 show that unsupervised contrastive training generally yields comparable or better acceptablity estimates for constituency analysis than the CoLA-based model. Experiments with different sets of pronouns for training and test application reveal the robustness of our approach: In particular, pretraining with a set of pronouns that is complementary to those pronouns used for constituency analysis yields good results as well. In general, training with a large set of pronouns and performing constituent analysis with a selection of pronouns works best. Focused contrastive training, i.e. selecting and marking positive and negative samples so that they are most helpful for decisions at analysis time has a consistently positive effect in all settings.

References

Steven Cao, Nikita Kitaev, and Dan Klein. 2020. Unsupervised parsing via constituency tests. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4798–4808, Online. Association for Computational Linguistics.

Danqi Chen and Christopher D Manning. 2014. A fast and accurate dependency parser using neural networks. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 740–750.

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.

Jingtao Ding, Fuli Feng, Xiangnan He, Guanghui Yu, Yong Li, and Depeng Jin. 2018. An improved sampler for bayesian personalized ranking by leveraging view data. In *Companion Proceedings of the The Web Conference 2018*, WWW '18, page 13–14, Republic and Canton of Geneva, CHE. International World Wide Web Conferences Steering Committee.

Chris Dyer, Miguel Ballesteros, Wang Ling, Austin Matthews, and Noah A Smith. 2015. Transition-based dependency parsing with stack long short-term memory. *arXiv preprint arXiv:1505.08075*.

Takuma Ebisu and Ryutaro Ichise. 2020. Generalized translation-based embedding of knowledge graph. *IEEE Transactions on Knowledge and Data Engineering*, 32(5):941–951.

Taeuk Kim and Sang-goo Lee. 2020. Multilingual zero-shot constituency parsing. *CoRR*, abs/2004.13805.

Nikita Kitaev and Dan Klein. 2018. Constituency parsing with a self-attentive encoder. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2676–2686, Melbourne, Australia. Association for Computational Linguistics.

Bowen Li, Taeuk Kim, Reinald Kim Amplayo, and Frank Keller. 2020a. Heads-up! unsupervised constituency parsing via self-attention heads. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, pages 409–424, Suzhou, China. Association for Computational Linguistics.

Jun Li, Yifan Cao, Jiong Cai, Yong Jiang, and Kewei Tu. 2020b. An empirical comparison of unsupervised constituency parsing methods. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3278–3283, Online. Association for Computational Linguistics.

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*.

Zhuang Ma and Michael Collins. 2018. Noise contrastive estimation and negative sampling for conditional models: Consistency and statistical efficiency. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3698–3707, Brussels, Belgium. Association for Computational Linguistics.

David Mareček and Rudolf Rosa. 2019. From balustrades to pierre vinken: Looking for syntax in transformer self-attentions. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 263–275, Florence, Italy. Association for Computational Linguistics.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'13, page 3111–3119, Red Hook, NY, USA. Curran Associates Inc.

Sebastian Riedel, Limin Yao, Andrew McCallum, and Benjamin M. Marlin. 2013. Relation extraction with matrix factorization and universal schemas. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 74–84, Atlanta, Georgia. 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. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 6000–6010, Red Hook, NY, USA. Curran Associates Inc.

Alex Warstadt, Amanpreet Singh, and Samuel R. Bowman. 2019. Neural network acceptability judgments. *Transactions of the Association for Computational Linguistics*, 7:625–641.

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*, volume 32. Curran Associates, Inc.

Yu Zhang, Houquan Zhou, and Zhenghua Li. 2020. Fast and accurate neural crf constituency parsing. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, pages 4046–4053. International Joint Conferences on Artificial Intelligence Organization. Main track.