Introducing Shallow Syntactic Information within the Graph-based Dependency Parsing

Nikolay Paev, Kiril Simov, Petya Osenova
IICT, Bulgarian Academy of Sciences

TLT 2024, Hamburg, Germany
06.12.2024

Plan of the Talk

- Introduction.
- Part 1: Graph-based parsing
- Part 2: Chunk grammars and MWE
- Part 3: Introducing a hybrid model
- Experimental setup
- Results
- Discussion

Introduction

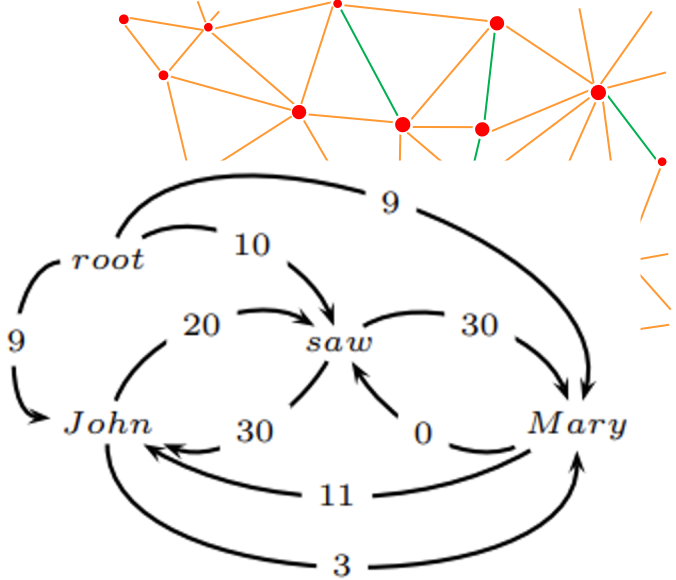
- A transformer-based architecture for dependency parsing is introduced
- It has been extended to accommodate some predefined shallow dependency information
- This information comes from two sources:
  - lexicons (valency relations and MWEs)
  - shallow grammars (non-recursive NPs and verbal complexes)

Syntax in BERT layers

- Manning et al. (2020) *Emergent linguistic structure in artificial neural networks trained by self-supervision*

- They optimize a metric which models the edge classification as a distance between vectors task
- They show that there is a syntax tree representation hidden in the BERT embeddings from different layers

Part 1: Graph-based Parsing

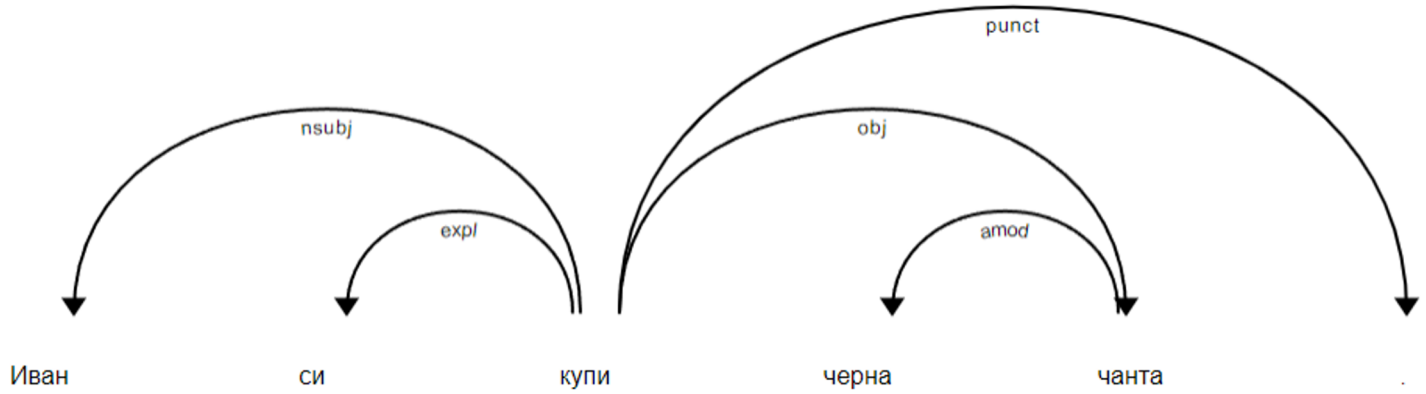


- UAS - assigning the heads to each dependant
  - scoring function on top of the encoder
    - similar to attention
    - produces probabilistic distribution over the possible heads of each dependant
  - end-to-end fine-tuning
  - a full graph is created over the tokens in the sentence with weights
  - a Maximum Spanning Tree is selected as a syntactic tree

- Labeling after UAS
  - classifier layer that classifies (dependant, head) -> label
  - end-to-end fine-tuning

Example scoring output

	ROOT	Иван	си	купи	черна	чанта	.
Иван	-inf	-2.92	-3.60	10.67	-11.72	-3.01	-16.77
си	-inf	-6.68	-8.91	5.46	-13.26	-8.31	-20.97
купи	22.69	-0.41	2.28	6.14	-13.78	-6.55	-18.69
черна	-inf	-9.32	-11.54	-6.28	-9.36	9.08	-20.77
чанта	-inf	-5.27	-5.46	10.78	-7.01	-0.96	-14.99
.	-inf	-0.34	0.13	22.06	-9.87	3.69	-25.23



Ivan bought himself a black bag.

UAS in Depth

- Scoring layer - dot product of word embeddings after separate linear projections
  - $h_i = Model(w_i)$ $q_i = QueryMatrix(h_i)$ $k_i = KeyMatrix(h_i)$
  - softmax(q_iK) - distribution over the possible heads of i-th word
- Full weighted graph (matrix - token count X token count)
  - QK
- Training with Cross Entropy Loss

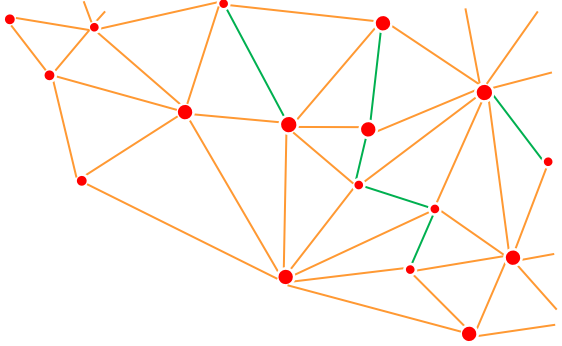
$$\text{Loss} = -\sum_{i=1}^{s}\sum_{k=0}^{s} y_{i,k}\log((\text{softmax}(q_iK))_k) \qquad y_{i,k} = \begin{cases} 1, & \text{if } w_k \text{ is the head of } w_i \\ 0, & \text{otherwise} \end{cases}$$

- Inference with argmax (greedy) or MST

Pre-trained encoders and Resources

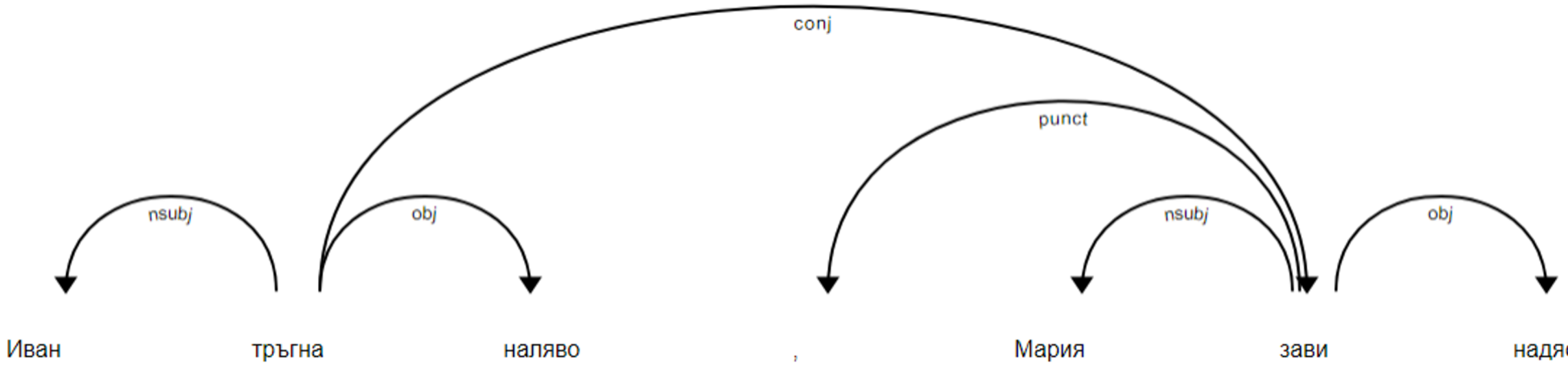
- Pre-training Dataset - Literature, News articles, Webpages, Wikipedia and other sources. Size - 20B tokens
- Pre-trained models used in the experiments:
  - BERT - base 109M
  - BERT - large 334M
- Syntactic Language Resources for Bulgarian
  - UD Bultreebank
  - MWE - complex pronouns, prepositions, conjunctions, phraseology (in process of creation)
  - Chunk grammars
  - Valency lexicon of Bulgarian (not used in these experiments)

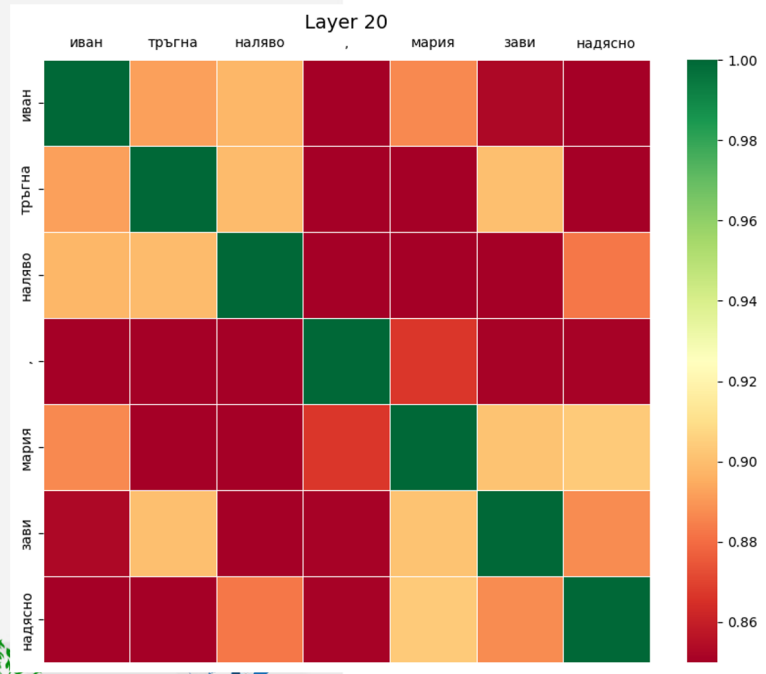
UAS (Basic Model)

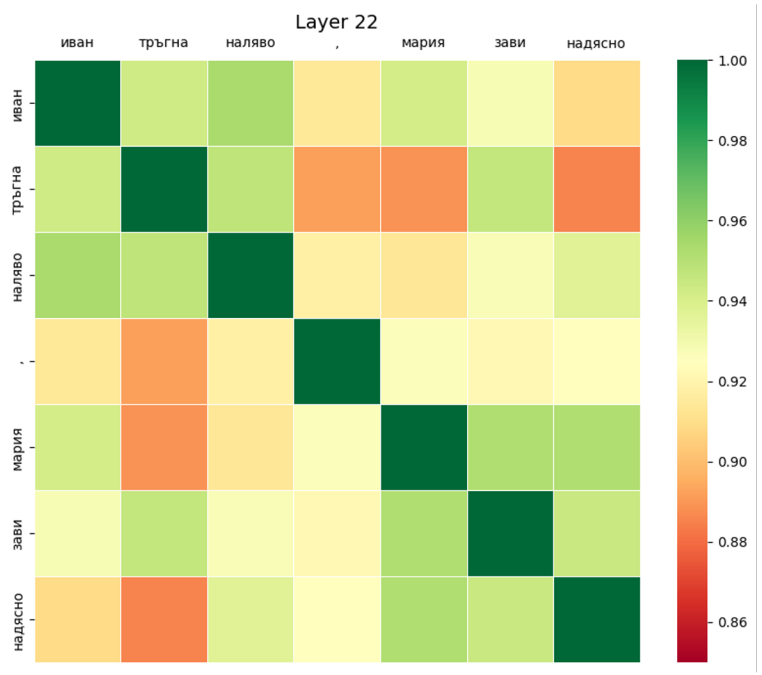


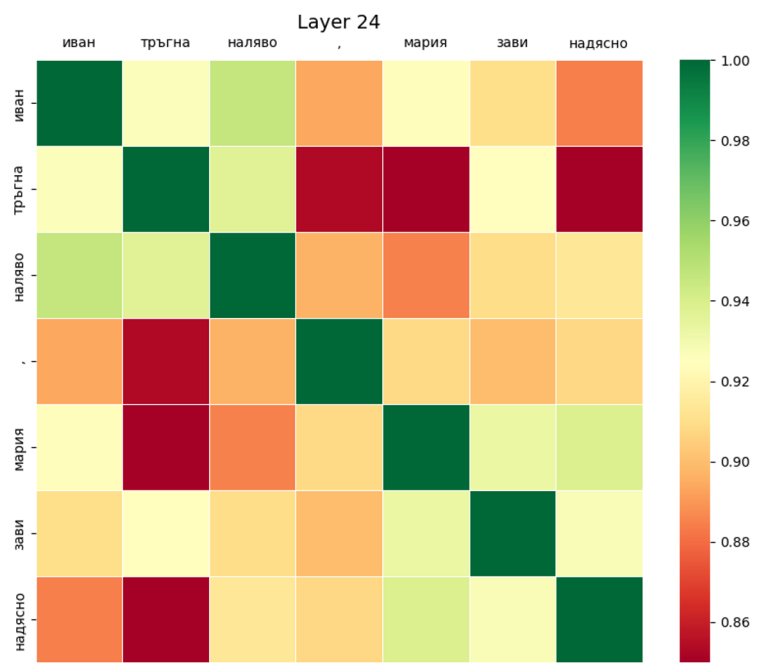
- Using MST only leads to small improvement
  - The greedy algorithm produces correct trees (no cycles) almost all the time 99+%
  - Decisions for the tree structure must have been made in the last layers of the fine-tuned encoder

- Next slide - Cosine similarity of word embeddings from different layers over an example sentence


conj
punct
nsubj
obj
nsubj
obj
Иван
тръгна
наляво
,
Мария
зави
надясно


Layer 20
иван
тръгна
наляво
,
мария
зави
надясно
1.00
0.98
0.96
0.94
0.92
0.90
0.88
0.86


Layer 22
иван
тръгна
наляво
,
мария
зави
надясно
1.00
0.98
0.96
0.94
0.92
0.90
0.88
0.86

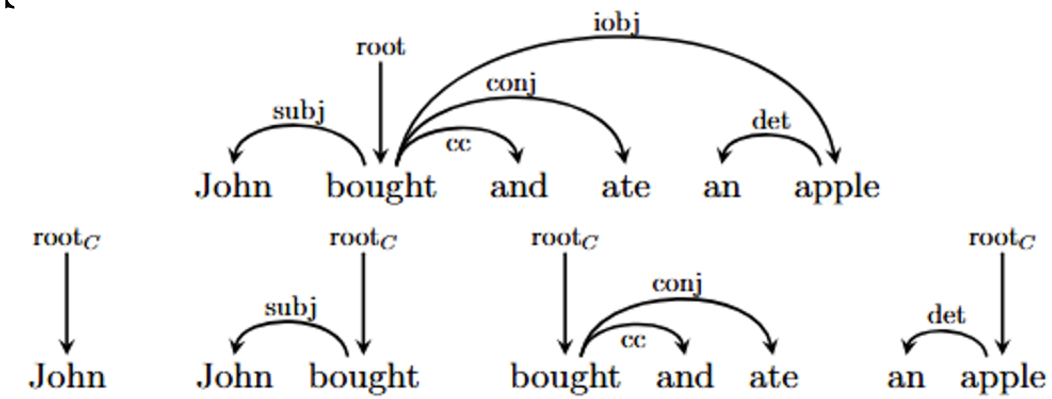

Layer 24
иван
тръгна
наляво
,
мария
зави
надясно
1.00
0.98
0.96
0.94
0.92
0.90
0.88
0.86

Part 2: MWE lexicon and chunk grammars

- Not all arcs need to be predicted
- The additional syntactic information is coming in form of arcs between tokens in the sentence (including root) with optional dependency labels
- Each arc between two tokens is used to help find the correct parse of the sentence

MWE Representation in the Lexicon

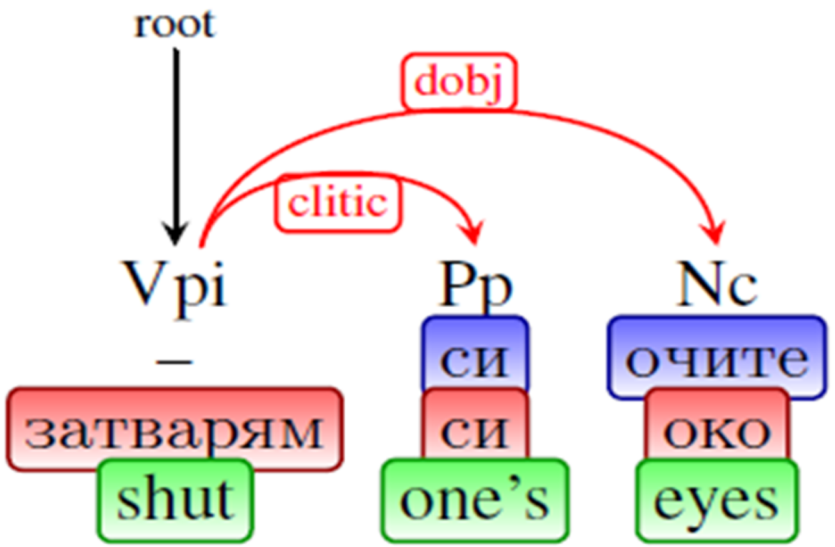
- We assume that each MWE is represented as a syntactic subtree in the lexicon (see Osenova and Simov, 2024). We call such a kind of subtree catena
- Catena example:



- The catena for a given MWE in the lexicon represents the information that is always presented in each realization
- The information from the lexicon representation could be used as "sure" information during the parsing

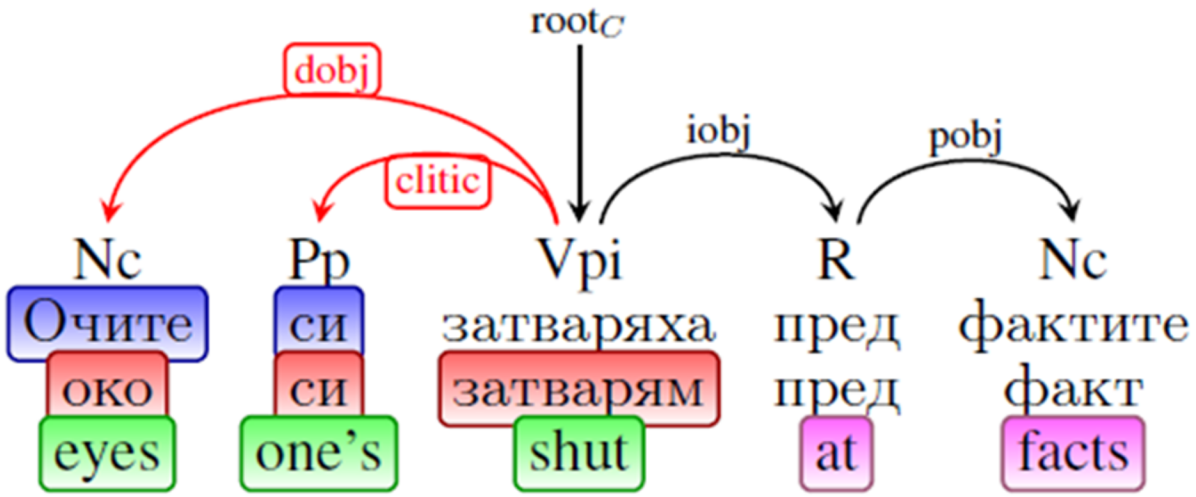
Representation of a Catena

- Each catena is a connected subtree of a dependency tree for a sentence
- The root of the subtree is mark up
- Each token contains information for the lemma, word form and morphosyntactic information

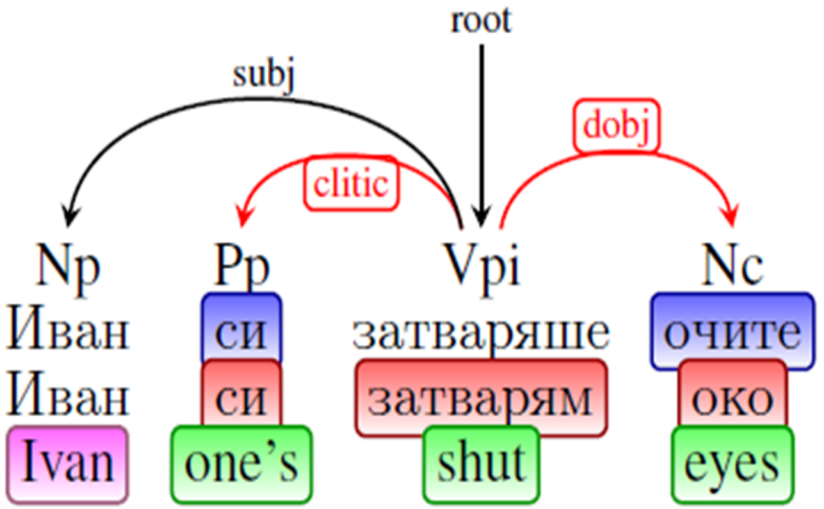


Catena Realization

Realization 1:

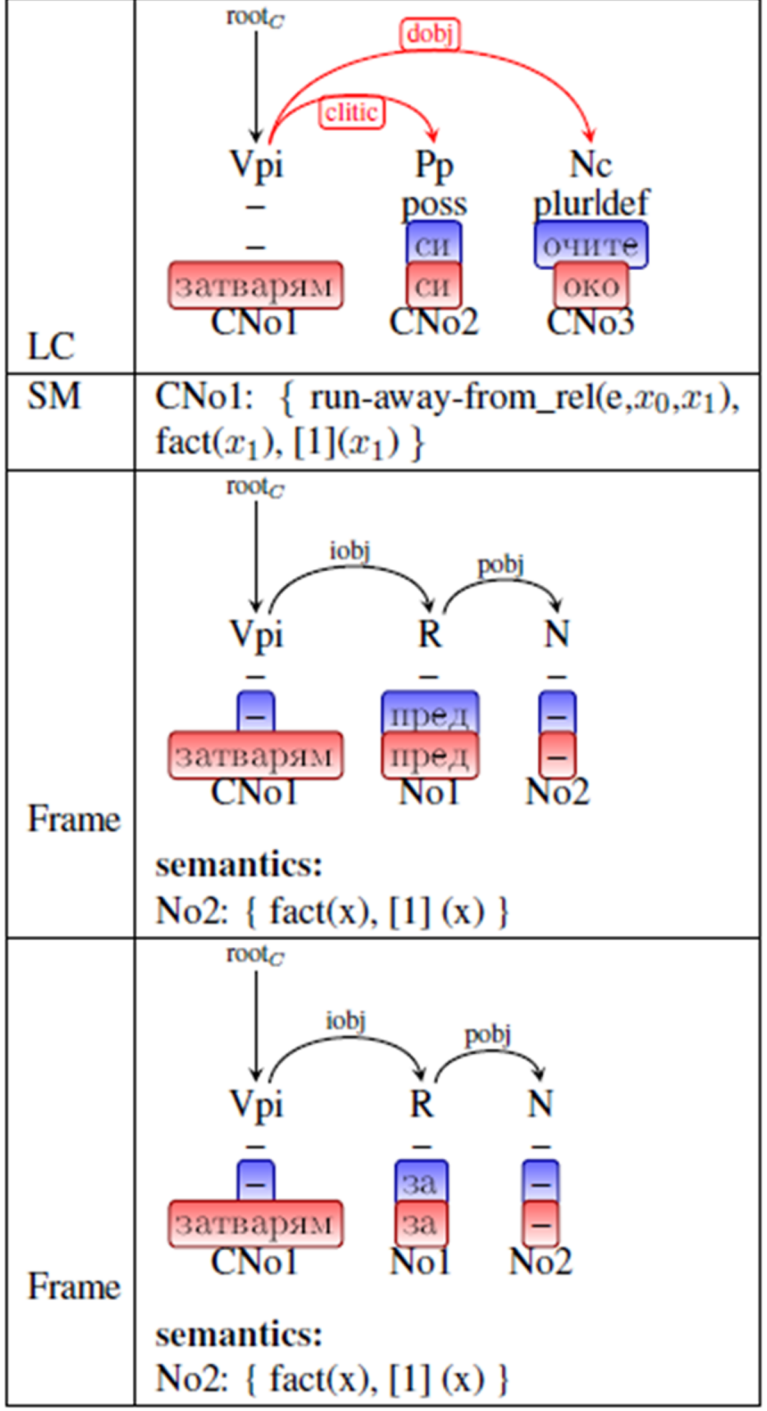


Realization 2:



Verbal MWE Lexical Entry

- The entry consists of:
  - lexical catena,
  - semantic restriction;
  - frames
  - frames semantic contribution
- In the experiments we are using the arcs from the lexical catena



Partial (Chunk) grammar

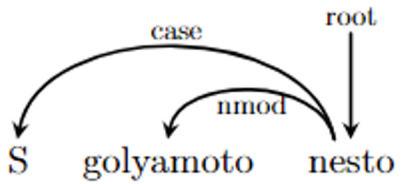
- During the creation of BulTreeBank project an extensive set of chunk grammars for Bulgarian were implemented
- Here is an example:

Rules:

NPns -> (An#|"Pd@@@sn), (Pneo-sn|Pfeo-sn)
PP -> R,N#

Analysis:

PP ->
R -> C
NP ->
Ansd -> голямото
Pfeo-sn -> нещо

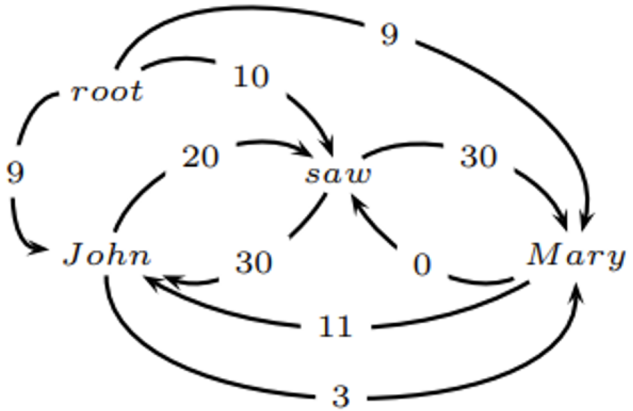


With the big thing

Part 3: Hybrid Model

- Not all head - dependent pairs need to be predicted

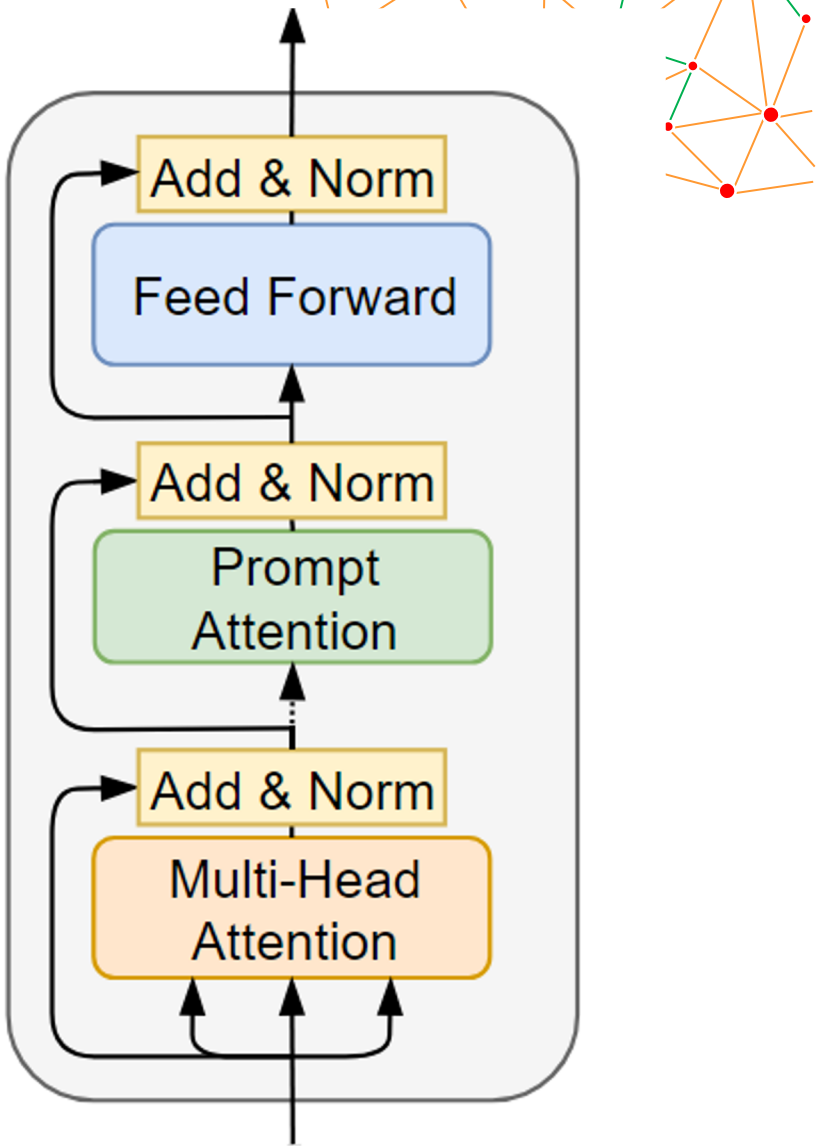
- First approach (Baseline):
  - the model predicts the weights,
  - correct them based on the deterministic knowledge and
  - then run MST



- Better approach:
  - incorporate “sure” information in the layers,
  - provide the deterministic knowledge to the model before it makes the predictions,
  - fine-tuning together with “sure” information

Prompt Attention

- Introducing a new sublayer in the encoder
- Adds the transformed vector of the predefined head to the vectors of the dependant
- Tries to tap into the presumed syntax tree representation hidden in the encoder layers



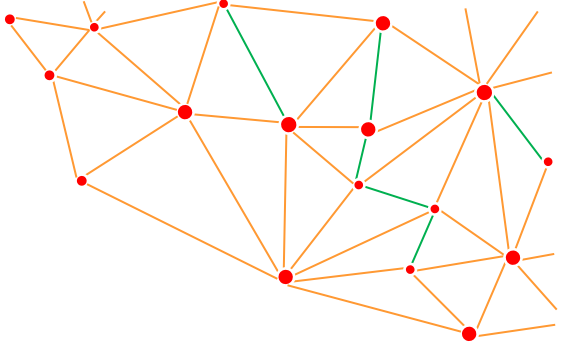
Prompt Attention

- Linear projection of the head
  - *Prompt(x)*
- Added to the dependant embeddings

$$O_{pa}(X) = \left(LN(w_i + \sum_{j=1}^{S} I(i,j) * Prompt(w_j))\right)_{w_i \in X}, \qquad I(i,j) = \begin{cases} 1, & \text{if } w_j \text{ is the predefined head of } w_i \\ 0, & \text{otherwise} \end{cases}$$

- Modifying only the last few layers
  - Many modified layers leads to too many new parameters

Training and testing setup



- Training with random set of predefined arcs leads to better generalization (as opposed to training with Chunk grammars and MWE lexicon predefined arcs)
  - more control over the ratio of the predefined arcs - too much or too little lead to worse performance
  - more diverse types of arcs

- Testing with Chunk grammars and MWE lexicon predefined arcs

Experiments and testing setup

- We compare to the MST model which incorporate prompts into the encoder but after inference is corrected with the deterministic knowledge

- The prompted model uses the prompts successfully and manages to generalize over them

Results for Bulgarian

Model	Training set	$T_P(udt_1)^{ChMWE}$ UAS	$T_P(udt_1)^{ChMWE}$ LAS	$T_P(udt_1)^{0}$ UAS	$T_P(udt_1)^{0}$ LAS	$T_P(udt_1)^{20}$ UAS	$T_P(udt_1)^{20}$ LAS
Prompted-0	$T_P(udt_1)^{0}$	0.9640	0.9361	***0.9615***	***0.9337***	0.9695	0.9410
Prompted-10	$T_P(udt_1)^{10}$	0.9655	0.9370	0.9626	0.9340	0.9690	0.9400
Prompted-20	$T_P(udt_1)^{20}$	0.9641	0.9360	0.9606	0.9324	0.9700	0.9411
Prompted-0-40	$T_P(udt_1)^{0+40}$	**0.9672**	**0.9392**	0.9640	0.9362	0.9718	0.9433
Prompted-ChMWE	$T_P(udt_1)^{ChMWE}$	0.9655	0.9374	0.9510	0.9231	0.8307	0.8665

$T_P(udt_i)$ - a subset of the treebank udt_i; $T_P(udt_i)^J$ - a subset of the treebank with J amount of predefined arcs; $T_P(udt_i)^{ChMWE}$ - a subset of the treebank with arcs from Chunk Grammars or MWE added to it; $T_P(udt_i)^{20}$ - a subset with 20% randomly selected arcs as predefined; $T_P(udt_i)^{0}$ - the treebank subsets without any predefined arcs.

Improvements in UAS (LAS)

	UAS	LAS
BERT Parser	96.15%	93.37%
BERT Parser + MWE and CG	96.40%	93.61%
Prompted BERT Parser with MWE and GGs	**96.72%**	**93.92%**

Prompting Works for English

- google bert large uncased - same number of non-embedding params
- gum tree bank - similar size tree bank
- no available chunk grammar and MWE lexicon so choosing random edges as a proof of concept
- prompted model is still better than baseline

Model	**Training set**	$T_P(GUM)^{20}$		$T_P(GUM)^0$	
		UAS	**LAS**	**UAS**	**LAS**
Corrected Argmax	$T_P(GUM^0)$	0.9436	0.9246	0.9299	0.9125
Corrected MST	$T_P(GUM^0)$	0.9447	0.9256	***0.9308***	***0.9133***
Prompted-10	$T_P(GUM)^{10}$	0.9467	0.9273	0.9321	0.9143
Prompted-20	$T_P(GUM)^{20}$	**0.9471**	**0.9280**	0.9310	0.9138

BulTreeBank CLaDA BG

Manual Evaluation

- The baseline makes errors
  - wrong head direction *(the subject of a copula depends on the coppola instead of the content word)*
  - wrong head selection *(NN construction with the first noun indicating quantity, the head is the first noun, but the model selected the second one)*
  - wrong head assignment *(the subject should be related to the main verb of a sentence but it was assigned to the modal verb instead)*
  - wrong root assignment *(in complex sentences, the baseline assigns the root relation to both verbs)*
  - wrong PP attachment *(instead of depending on the noun, the head of the PP is made dependant on the verb)*
- The best model makes errors
  - wrong head direction *(similar)*
  - wrong head assignment *(similar)*
  - wrong PP attachment *(similar)*
  - wrong non-PP attachment *(the adverb is adjacent to the preceding noun but has to be attached to the following verb, but it was wrongly attached to the noun)*

Conclusions from Manual Evaluation

- The best is not monotonically better than the baseline model
- Addressing this problem we plan to:
  - improve the prompt attention layer by including more linguistic information such as higher order arc information, grammatical features, shallow semantic information;
  - extent the treebank with new sentences selected using some active learning procedure;
  - improve the shallow grammar and the coverage of the MWE lexicon as well as the related algorithms for their better prediction and consequent recognition in text

Conclusion and Discussion

- The paper introduced a Hybrid Parsing model which incorporates deterministic information in the probabilistic model
- The model benefits even when it is used without the deterministic information
- The findings for Bulgarian remain consistent in English
- In future we plan to do experiments with models in which more linguistic information will be added
- Many of the errors identified by the manual investigation are related to rare phenomena in text, thus we need to extend the treebank

References

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. "Attention is All You Need." In Advances in Neural Information Processing Systems, 5998-6008, 2017.

Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. "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)

Manning, C. D., Clark, K., Hewitt, J., Khandelwal, U., & Levy, O. (n.d.). Emergent linguistic structure in artificial neural networks trained by self-supervision. Stanford University, Facebook Artificial Intelligence Research.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, Alexander M. Rush. 2019. HuggingFace's Transformers: State-of-the-art Natural Language Processing. https://arxiv.org/abs/1910.03771