Investigating Embeddings of Czech Nouns by Case

Jacob Dunefsky
Yale University / New Haven, CT
jacob.dunefsky@yale.edu

1 Introduction

Grammatical case, which can be defined as "the alternation in the form of a nominal or adjectival constituent based on its function" (Polinsky and Preminger, 2014), is a common feature in many languages. For example, consider the Latin sentence *Puella puerum amat.* (girl-NOM boy-ACC love-3.SG) "The girl loves the boy". In this sentence, *puella* "girl" is the subject, the one doing the loving; *puerum* "boy" is the object, the one receiving the loving. Now, consider the situation in which the roles are reversed: *Puellam puer amat.* (girl-ACC boy-NOM love-3.SG) "The boy loves the girl". Here, the order of the words is the same, unlike in the English translation; what changes is the form of the nouns: *puella* becomes *puellam* and *puerum* becomes *puer*. This is the essence of grammatical case.

In many languages containing grammatical case, the precise way in which a noun is altered in order to express a specific case differs between nouns. For instance, returning to Latin, the genitive case of *campus* "field" is given by *campi*, but the genitive case of *corpus* "body" is given by *corporis*. In the first example, the *-us* drops away and is replaced by *-i*, but in the second case, the same *-us* ending is replaced by *-oris*. Additionally, there are many instances in which words in different cases have the same form. Using another Latin example, *ignis* "fire" is both a valid nominative and genitive form. And yet, it is possible to make statements such as "*ignis* is in the genitive in the sentence *Ignis lucem vidit puer* (fire-GEN light-ACC see-3.SG boy-NOM) 'The boy sees the light of the fire'". Clearly, this means that grammatical case is not a mere property of morphology or syntax, but something deeper, with roots in semantics. We might then ask ourselves: to what extent are these categories constructed by grammarians, rather than understood by the people speaking the language? Would your average ancient Roman consider *ignis* in the genitive to be something different from *ignis* in the nominative? (Similar questions are discussed in FREDE (1994).)

Nowadays, large language models such as Transformers have shown exceeding promise in a wide variety of natural language tasks, producing near-human-level output in many situations (Brown et al., 2020). Given that we have the ability to poke and prod the internals of such powerful models, it is thus a natural question to ask: can we measure the degree to which language models understand grammatical case?

2 Related work

The most relevant prior work is that of Kawasaki and Kimura (2018), which uses an MLP to determine the "deep case" of nouns in Japanese sentences. Deep case, defined in contrast to "surface case" in Bruce (1975), acts on a "semantic level" rather than a syntactic one. This is particularly pertinent to the Japanese language, because surface case marking in Japanese is completely regular. Any noun can be put into the surface accusative form by appending the particle "wo" to the noun; any noun can be put into a dative form by appending "ni" (Aoyagi, 1998). But the meaning of these two surface cases differs depending on context. For example, "wo" marks the direct object of transitive verbs, but the medium through which motion is undergone with verbs of movement. Additionally, "ni" marks the indirect object of transitive verbs, but the agent of certain passive and intransitive verbs, and the target of verbs of motion.

Similarly to Kawasaki and Kimura (2018), our work attempts to measure the degree to which a neural network can model the semantics of case. However, the language used in our experiments,

Czech, is unlike Japanese in that case marking is not regular. Instead, similarly to Latin, Sanskrit, Russian, Icelandic, and other Indo-European languages, Czech nouns are marked for case following various declension paradigms. For example, the word *rok* "year-NOM" appears in the dative as *roku* "year-DAT", but the word *žena* "woman-NOM" appears in the dative as *ženě* "woman-DAT". Thus, rather than try to predict deep case from a given regular surface form, our work attempts to measure whether a neural network associates the same surface cases with the same semantics.

3 Approach

3.1 Goal

Generally, in attempting to determine the degree to which a model such as a Transformer "understands" the meaning of different noun cases, a good beginning would be to examine the model's representations of words and the relationships between them. One popular method of doing so is to consider a relationship between words to be represented by the vector difference of their embeddings. This approach has been popular since the seminal work of Mikolov et al. (2013), in which word2vec, a fast word embedding model, was introduced. Using the notation [[x]] to denote the embedding of the word x, the authors explained that "To find a word that is similar to *small* in the same sense as *biggest* is similar to *big*", the vector $X = [[\text{biggest}]] - [[\text{big}]] + [[\text{small}]]$ is computed. Then, "we search in the vector space for the word closest to X measured by cosine distance, and use it as the answer to the question". This approach makes use of the fact that the *biggest-big* relationship can be represented by the vector difference $[[\text{biggest}]] - [[\text{big}]]$. The authors use this approach to answer a wide variety of analogies, such as "France : Paris :: Italy : Rome", "Einstein : scientist :: Mozart : violinist", and "Microsoft : Ballmer :: Apple :: Jobs".

Now, consider an inverse task: we have a predefined relationship – e.g., the *country-capital city* relationship of which an example was given above. We want to measure the strength of this relationship, the degree to which this relationship is meaningful. For instance, the *country-capital city* relationship is clearly meaningful to the embedding model, since the difference vectors representing *individual examples of this relationship*, like $d_1 = [[\text{France}]] - [[\text{Paris}]]$ and $d_2 = [[\text{Italy}]] - [[\text{Rome}]]$, are approximately the same. Now, instead of considering the difference vectors themselves, let us consider the directions in which they point. Indeed, researchers such as Fournier et al. (2020) argue that the overall goal of these word analogies is to measure the degree to which there exists "the presence of a regular direction encoding relations such as *capital-of*: *France–Paris*, *China–Beijing*", arguing that word analogies are an improper tool for measuring the semantic understanding of a word embedding model to the extent that factors other than the presence of this regular direction exist. Thus, as a proxy for measuring the degree to which such a relationship is meaningful, we could measure the degree to which difference vectors point in the same direction.

3.2 Mean angle deviation

To measure how much any set of vectors point in the same direction, let us define the **mean angle deviation** (MAD) as follows. Let X be an arbitrary set of vectors. Then, if $\overline{X}$ denotes the mean vector of X, then mean angle deviation can be defined as

$$MAD(X) = \frac{1}{|X|} \sum_{x \in X} \arccos\left(\frac{x \cdot \overline{X}}{\|\overline{X}\|\|x\|}\right)$$

In words, this is the mean angle between each vector in X and the mean of X. Intuitively, if there is a "regular direction" in which the vectors of X point, then this measure tells how far from that direction one should expect a random vector of X to be.

There are other measures of circular statistical dispersion, such as the "circular standard deviation". This measure, however, was chosen because it is graphically intuitive and immediately interpretable.

Note that in general, finding the mean of a set of angles is not as simple as summing the angles and dividing by the number of angles. However, because all angles involved are less than or equal to π, this is not of any concern.

When working with the MAD in high-dimensional space, there is an important piece of informal intuition to keep in mind: the higher the dimension, the closer the MAD will be to $\pi/2$. This is due to the same properties of high-dimensional space that are responsible for the Johnson-Lindenstrauss lemma – namely, that the higher the dimension, the more likely any two vectors are to be orthogonal in general.

Also, note that for the sake of computational efficiency, the MAD uses the mean vector $\overline{X}$ rather than the mean of normalized vectors. A quick empirical comparison found that there was less than one degree of difference between the two metrics for all measurements when using the fastText model (see Experiments for more details on this model), so the MAD was computed as explained.

Finally, note that when $\|x\| = 0$ (or is close enough to 0 for the purposes of floating point division), it is simply ignored, and that datapoint is not counted towards the MAD. This is theoretically justifiable because the presence of that datapoint does not affect the direction in which the mean vector $\overline{X}$ points. Intuitively, adding zero vectors doesn't change whether or not the vectors in the set point in the same direction.

3.3 Case difference vector sets

Having defined the MAD, it can be applied to sets of case difference vectors. Let $C(w)$ denote the form of w in the case C. For instance, if the language is Latin and w is *rex* "king-NOM", then $\text{dat}(w) =$ *regi* "king-DAT". Then, for cases C_1 and C_2, define the set

$$D_{C_1,C_2} = \{[[C_1(w)]]-[[C_2(w)]] \mid w \text{ in our dataset}\}$$

to be the **case difference set** of C_1 and C_2. This is the set of all individual examples of the relationship between C_1 and C_2, as explained in Section 3.1. $MAD(D_{C_1,C_2})$ thus measures the degree to which the C_1-C_2 relationship is meaningful.

Additionally, define the set

$$U_{C_1} = \{[[C_1(w)]]-[[C_r(w)]] \mid w \text{ is in our dataset}\}$$

where C_r is a random case chosen uniformly from the cases that are not C_1. This set is the **case uniqueness set** of C_1. One way of thinking about $MAD(U_{C_1})$ is that it can be used to determine the degree to which the model encodes information about a noun beyond its case (see Section 6.2.2).

Finally, define the set

$$R_{C_1} = \{[[C_1(w)]] - [[r]] \mid w \text{ is in our dataset}\}$$

where r is a completely random word form. This set is the **baseline set** of C_1. $MAD(R_{C_1})$ is used, as the name suggests, as a baseline; $MAD(D_{C_1,C_2})$ and $MAD(U_{C_1})$ should both be lower than $MAD(R_{C_1})$ – particularly the former. If $MAD(D_{C_1,C_2})$ is around the same as $MAD(R_{C_1})$, then it means that the model's understanding of the relationship between case C_1 and C_2 is not much better than the model's understanding of the relationship between C_1 and completely random words, random words which are not even derived from the same root as the words in C_1.

4 Experiments

Having defined and motivated these quantities, we now want to measure them, given a language with irregular case marking and a model which produces embeddings of words from that language. The language chosen for our experiments was Czech, due to its irregular case marking (see Section 2, in which Japanese and Czech are contrasted).

4.1 Dataset

A dataset of Czech noun forms, grouped by case, was constructed. First, the 2000 most frequent Czech nouns were scraped from a frequency list computed from the SYN2015 corpus (Křen et al., 2016). Then, indeclinable forms such as abbreviations (e.g. "EU", short for "European Union") were filtered. All of the remaining nouns were given in the nominative singular form, so their plural and non-nominative forms were found by scraping Wiktionary, which maintains declension tables for Czech nouns. However, many feminine nouns didn't have declensions listed on Wiktionary, because feminine nouns have more regular declension patterns. Thus, when no declension was found, the scraper supplied them automatically from a table, depending on the final vowel of the noun. In the end, there were 1574 nouns for each of the seven Czech cases. However, this does not come out to $1574 \times 7 = 11,018$ distinct forms, because there is substantial overlap between certain cases in certain declension patterns. For instance, the nominative and accusative cases of masculine inanimate nouns are the same.

Note that for Transformer models, only the 400 most common nouns, in each case were used, due to computational limitations. With fastText, the 1000 most common nouns were used.

4.2 Models

Three models were used in our experiments: fastText precomputed embedding vectors (Grave et al., 2018), the RobeCzech pretrained Transformer (Straka et al., 2021), and the RobeCzech Transformer finetuned on a case identification task.

The fastText model was "trained using CBOW with position-weights, in dimension 300, with character n-grams of length 5, a window of size 5 and 10 negatives".

RobeCzech is a BERT model trained with RoBERTa, a "robustly optimized BERT pretraining approach". RobeCzech was "trained solely on Czech data" coming from the SYNv4 corpus, the Czes corpus, the "Czech part of the web corpus W2C", and "plain texts extracted from Czech Wikipedia dump 20201020".

4.2.1 Finetuned Transformer

The finetuned transformer was trained on a case prediction task adapted from the Czech noun form dataset created in this experiment. Each word form was labeled with a seven-dimensional vector, with a 1 in position i if the word form appeared in case i, and a 0 otherwise. For example, the word *rok* "year" would have label $[1, 0, 0, 1, 0, 0, 0]$, with ones in the places representing the nominative and accusative case, because the nominative and accusative form of *rok* are the same: *rok*. In contrast, *roku* would have vector $[0, 0, 1, 0, 0, 0, 0]$, with a one in the place representing the dative case, because it only appears in the dative case. In total, the dataset contained 6886 different word forms.

The decoder of the pretrained model was replaced with a linear layer. The model was then trained with cross-entropy loss for 3000 steps on a randomly chosen 80% of the dataset. The optimization hyperparameters were the default used in HuggingFace's Trainer class (Wolf et al., 2020).

4.2.2 Working with Transformer embeddings

It is worth noting that both the finetuned Transformer and RobeCzech have a hidden vector size of 768, more than twice as large as fastText.

For the Transformer models, embeddings were calculated for each hidden layer by taking the mean embedding over all tokens.

5 Results

For all C_1 and $C_2 \in \{$ nominative, genitive, dative, accusative, vocative, locative, instrumental $\}$, the values

$$DBB(C_1, C_2) = MAD(R_{C_1}) - MAD(D_{C_1,C_2})$$

and

$$DBB(C_1, \text{all}) = MAD(R_{C_1}) - MAD(U_{C_1})$$

were calculated. Tables 1, 2, and 3 give these values for fastText, RobeCzech, and the finetuned model as "degrees below the baseline" (DBB); a higher number is thus better. To find $DBB(C_1, C_2)$, go to row C_1 and column C_2. For each row, the greatest value was set in bold typeface. Additionally, for the two Transformer models rather than display the DBB for the embeddings calculated by every layer of the Transformer, the maximum DBB over all layers is given, along with the layer at which this maximum was achieved. Thus, as an example, when the vocative-dative cell in Table 2 reads "10.37 at 7", it means that the embeddings at layer 7 of the RobeCzech model produced the greatest DBB, which was 10.37 degrees below the baseline.

6 Discussion

6.1 fastText embeddings vs RobeCzech embeddings

Looking at the tables, the fastText embeddings and RobeCzech embeddings are somewhat similar in magnitude, although the fastText embeddings tend to yield higher DBB values than the RobeCzech embeddings. However, the finetuned model yields higher DBB values for every single relationship than both other models, and in some cases, these values are far higher. This is to be expected: the finetuned model is trained to predict the case of a word, so it will produce embeddings that more greatly reflect these case relationships.

6.2 A theoretical perfect case predictor vs. the finetuned model

For the purpose of comparison, let us investigate how a perfect case predictor model would fare under the metrics used in this experiment. Let us ignore all ambiguous forms (e.g. word forms that could be more than one case) for the sake of argument. Then, we could imagine the model embedding each word according to its case and solely its case, in a subspace of embedding space isomorphic to $\mathbb{R}^7$. Intuitively, we could think of each word's embedding as being its label, a one-hot vector.

6.2.1 DBB of case difference sets

It immediately follows from the above that the MAD of each case difference set would be 0.

Let us calculate now what the baseline set would be. Without loss of generality, consider the nominative case, with label $[1, 0, 0, 0, 0, 0, 0]$. The various vectors in R_{nom} would thus be 0,

$-[0,1,0,0,0,0,0]$, $-[0,1,0,0,0,0,0]$, and so on, in equal proportion. Thus, the mean vector $\overline{R_{\text{nom}}}$ would be proportional to $-[0,1,1,1,1,1,1]$. Now, for purposes of calculating the MAD, all the zero vectors are removed from consideration. By symmetry, the rest of the vectors have the same angle with the mean vector: approximately 65.905 degrees. This is the MAD. Thus, the DBB of each case difference set, in this absolute theoretical limit, would be 65.905 degrees.

The highest DBB of a case difference in the finetuned Transformer is 42.35 DBB; thus, the finetuned Transformer made it approximately 64.3% of the way to the theoretical limit.

6.2.2 DBB of uniqueness sets

It is also worth considering the DBB of $MAD(U_{\text{nom}})$. As it turns out, U_{nom} is just R_{nom}. Thus, the DBB of $MAD(U_{\text{nom}})$ is just 0! Intuitively, this is a consequence of the model throwing away the semantic information associated with each word form beyond its case. As such, from the model's perspective, the other forms of the same noun are just as foreign from the noun as completely random words in different cases.

In contrast, the DBB of the uniqueness sets of the finetuned Transformer are actually *higher* than those of the two other models. This provides some evidence that finetuned Transformer has not fallen into the same failure mode as the perfect case predictor. However, much more testing would be required to ensure that the finetuned Transformer still retains its ability to model Czech in general (see Section 7); other failure modes are certainly possible.

6.3 Transformer layers and case

Among the Transformer models, there exist clear patterns in which case relationships were maximally observed at which layers. Consider the RobeCzech model. Case relationships with the nominative are primarily maximally observed in the layer 11 embeddings. Case relationships with the accusative are also primarily maximally observed at this layer. However, case relationships with the vocative are primarily observed maximally at layer 7, and case relationships with the instrumental appear primarily maximally observed in layers 6 and 7. Interestingly, in the RobeCzech model, the final embedding layer, layer 12, does not produce embeddings that display maximum DBB even once. This is despite the frequent presence of layer 11. This seems to imply that between layers 11 and 12, most case information is discarded in order to produce the embedding corresponding to the output token.

In clear contrast, in the finetuned model, layer 12 embeddings yield maximum DBB values 22 times. However, when we consider that the finetuning objective of the finetuned model is to predict case, this makes clear sense: having case information available in the embedding layer closest to the output layer makes the model's task easier.

In general, the number of times that each layer's embeddings provided the maximum DBB is listed in Table 4 for RobeCzech and Table 5 for the finetuned model.

6.4 DBB and case

Looking at which cases participated in the relationships with the highest DBB can reveal interesting information about the semantics of Czech cases. In both the fastText and finetuned Transformer embeddings, the instrumental case appeared the majority of times as the case with the highest DBB. This means that when the baseline of the non-instrumental case is taken into account, the relationship with instrumental case tends to be strongest. One possible interpretation of this data is that the instrumental case is, in some semantic sense, the most distinguishable case, or the most unique case.

In the RobeCzech model, however, this is not the case. The nominative case appears three times as the case with the highest DBB, followed by the instrumental with two appearances, and the locative and dative with one appearance. Further investigation might explain why the situation is different for RobeCzech when compared to the other two models.

7 Future work

There are a number of directions in which this work could be continued. For example, more work could be done with the Transformer finetuned to predict case. The performance of this Transformer on common downstream NLP tasks could be evaluated and then compared to the performance of the original model, in order to see whether the process of learning explicit case information for each word caused the model to perform better or worse in real linguistic tasks.

Additionally, metrics other than mean angle deviation could be used to measure the strength of

the case relationships. Due to computational limitations, the computation of word analogies *a-la*-word2vec was not performed. However, there are many places in which the codebase used could be made vastly more performant. With performance improved, measures such as reciprocal rank gain (Finley et al., 2017) can be used to evaluate the degree to which case analogies are learned.

References

Hiroshi Aoyagi. 1998. *On the nature of particles in Japanese and its theoretical implications*. University of Southern California.

Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*.

Bertram Bruce. 1975. Case systems for natural language. *Artificial Intelligence*, 6(4):327–360.

Gregory Finley, Stephanie Farmer, and Serguei Pakhomov. 2017. What analogies reveal about word vectors and their compositionality. In *Proceedings of the 6th Joint Conference on Lexical and Computational Semantics (*SEM 2017)*, pages 1–11, Vancouver, Canada. Association for Computational Linguistics.

Louis Fournier, Emmanuel Dupoux, and Ewan Dunbar. 2020. Analogies minus analogy test: measuring regularities in word embeddings.

MICHAEL FREDE. 1994. The stoic notion of a grammatical case. *Bulletin of the Institute of Classical Studies*, 39:13–24.

Edouard Grave, Piotr Bojanowski, Prakhar Gupta, Armand Joulin, and Tomas Mikolov. 2018. Learning word vectors for 157 languages. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*.

Takumi Kawasaki and Masaomi Kimura. 2018. Deep case identification using word embedding. *International Journal of Computer Theory and Engineering*, 10(6).

Michal Křen, Václav Cvrček, Tomáš Čapka, Anna Čermáková, Milena Hnátková, Lucie Chlumská, Tomáš Jelínek, Dominika Kováříková, Vladimír Petkevič, Pavel Procházka, Hana Skoumalová, Michal Škrabal, Petr Truneček, Pavel Vondřička, and Adrian Jan Zasina. 2016. SYN2015: Representative corpus of contemporary written Czech. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 2522–2528, Portorož, Slovenia. European Language Resources Association (ELRA).

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space.

Maria Polinsky and Omer Preminger. 2014. Case and grammatical relations. In *The Routledge handbook of syntax*, pages 168–184. Routledge.

Milan Straka, Jakub Náplava, Jana Straková, and David Samuel. 2021. Robeczech: Czech roberta, a monolingual contextualized language representation model. *Lecture Notes in Computer Science*, page 197–209.

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, and Alexander M. Rush. 2020. Huggingface's transformers: State-of-the-art natural language processing.

	nom	gen	dat	acc	voc	loc	ins	all
nom	—-	13.76	13.67	**22.34**	8.97	13.39	17.73	9.92
gen	15.04	—-	17.51	15.27	11.00	17.09	**19.73**	9.45
dat	15.55	18.11	—-	15.23	10.38	17.25	**19.03**	10.05
acc	**23.17**	14.81	14.19	—-	8.87	13.96	17.80	9.13
voc	12.85	13.60	12.38	11.92	—-	12.15	**18.02**	5.07
loc	15.44	17.85	17.42	15.18	10.32	—-	**19.06**	9.89
ins	13.70	**14.42**	13.13	12.95	10.12	12.99	—-	10.38

Table 1: DBB for fastText embeddings

—-	nom	gen	dat	acc	voc	loc	ins	all
nom	—-	7.24 at 2	10.28 at 2	15.20 at 2	11.41 at 7	9.82 at 2	**16.75** at 6	8.56 at 2
gen	11.67 at 11	—-	11.83 at 3	6.48 at 11	11.68 at 7	10.95 at 3	**18.16** at 7	5.43 at 4
dat	16.06 at 11	11.90 at 3	—-	12.13 at 11	10.06 at 7	**22.47** at 11	17.92 at 7	7.30 at 11
acc	**18.05** at 2	5.42 at 3	6.54 at 2	—-	7.82 at 7	6.41 at 2	16.10 at 6	3.02 at 4
voc	**19.39** at 11	13.10 at 7	10.37 at 7	12.63 at 11	—-	10.24 at 8	17.17 at 7	9.02 at 7
loc	15.24 at 11	11.20 at 3	**22.81** at 11	11.33 at 11	10.01 at 7	—-	17.85 at 7	6.61 at 11
ins	**17.25** at 7	13.54 at 7	12.47 at 4	13.79 at 7	11.27 at 4	12.20 at 4	—-	12.04 at 7

Table 2: Maximum DBB and layer of maximum for RobeCzech

—-	nom	gen	dat	acc	voc	loc	ins	all
nom	—-	24.92 at 12	24.62 at 6	36.92 at 9	30.84 at 12	24.56 at 6	**37.74** at 6	15.18 at 5
gen	29.56 at 12	—-	36.49 at 12	27.42 at 12	34.59 at 12	35.61 at 12	**39.39** at 6	13.97 at 6
dat	28.79 at 12	36.76 at 12	—-	29.31 at 8	35.40 at 12	42.00 at 12	**42.12** at 7	15.58 at 6
acc	39.16 at 7	25.91 at 6	27.19 at 6	—-	27.36 at 12	27.17 at 6	**42.35** at 7	13.27 at 5
voc	32.71 at 12	31.82 at 12	32.35 at 12	27.11 at 11	—-	32.71 at 12	**38.74** at 7	12.27 at 12
loc	28.85 at 12	36.24 at 12	**42.36** at 12	29.45 at 8	36.11 at 12	—-	42.31 at 7	16.02 at 6
ins	20.73 at 5	**20.76** at 5	20.44 at 5	20.17 at 5	19.94 at 12	20.10 at 5	—-	16.03 at 4

Table 3: Maximum DBB and layer of maximum for finetuned Transformer

Layer	Frequency
7	16
11	12
2	8
3	5
4	5
6	2
8	1

Table 4: Frequency of layer of maxmimum for RobeCzech

Layer	Frequency
12	22
6	10
5	7
7	5
8	2
9	1
11	1
4	1

Table 5: Frequency of layer of maxmimum for the finetuned model