Linguistic Model Representation of Morphological Case Within and Across Languages with Declension

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

Ben Christensen[*]
Yale University / New Haven, CT
b.christensen@yale.edu

Abstract

Declension, or the changing in the structure of a word given its grammatical case, is a key facet in many of the world's languages. While languages such as English depend primarily on prepositions in order to change the relations of nouns with respect to their environment, other languages absorb these modifiers into the structure of the noun in question, thus changing the morphology. In such languages, this paper investigates the extent to which language models develop a sense of how cases within a single language relate to each other. Additionally, this paper explores the extent to which a multilingual language model is able to develop a cross-linguistic notion of grammatical cases, and the confounding effects of declensive diversity on the implemented mapping task.

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 videt 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?

In particular, given a language model which produces embeddings of words, we would like to ask the following two research questions:

1. Is there a **relationship between** word embeddings **in different cases within one language**?
2. Is there a **relationship between** word embed-

[*]Equal contribution

dings in **different cases across languages** in multilingual models?

Using Transformers, we achieve promising and informative conclusions for both of these questions.

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 (Question 1)

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 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 dat(w) = *regi* "king-DAT". Then, for cases C_1 and C_2, define the set

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

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 (Question 1)

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 (Question 1)

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. The higher average DBB of the fastText embeddings could potentially be explained by the much smaller dimension of the fastText embedding vectors; as explained earlier, one would expect the MAD of vectors in a lower-dimensional space to be lower. However, the DBB is calculated by taking into account the MAD of the baseline set, which raises the question of whether or not this also accounts for the lower dimensionality of the fastText vectors.

Regardless of this, 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 12); 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.

Overall, the results obtained indicate that within a single language with grammatical case, fastText embeddings and Transformer models do pick up on the relationships between cases. Having concluded this, it is now worth turning our attention to the question of whether relationships exist between cases *across different languages*.

7 Goal: Modeling cross-linguistic relationships (Question 2)

Our second question strives to discover if language models contain generalized knowledge of morphological cases across languages. Alternatively stated, we now investigate the extent to which a multilingual language model is able to agree on the concept of linguistic case across different languages.

To judge the model's ability to latently represent cases in different languages, we use a linear transformation.

Define h to be the embedding size. Let $x_{c_i} \in \mathbb{R}^h$ and $y_{c_i} \in \mathbb{R}^h$ denote the average embeddings of all words declined by case c_i, in languages x and y respectively. Assuming x and y share n distinct morphological cases, construct matrices $X = [x_{c_1} \cdots x_{c_n}]$ and $Y = [y_{c_1} \cdots y_{c_n}]$, and $X, Y \in \mathbb{R}^{h \times n}$. There exist matrices $A, B \in \mathbb{R}^{n \times n}$ such that $Y = XA$ and $X = BY$ lead to least-squares linear transformations. For each pair of languages, a lower error between the target matrix and transformation matrix product indicates that the linear transformation was more precise. In this case, it is clear that the model represented the absolute relationship between cases $c_1, \cdots, c_n$ in a congruent fashion for different languages, computable as a linear map. Thus, for each Y and X, we solve for A, B using least squares. A becomes

$$(X^T X)^{-1} X^T Y$$

and likewise for B with X and Y reversed. Wherever "present" is tabulated in Table 6 indicates that an average embedding vector in the form of $x_{c_i} \in \mathbb{R}^h$ will be computed for that language x and case c_i. Consider the following loss function:

$$\ell = \frac{1}{2}(||Y - XA||^2 + ||X - YB||^2)$$

To assess accuracy, we use this average least-squares function. Since the choices of Y and X are completely arbitrary (e.g. the relationship between Russian and Finnish can be computed by mapping Russian to Finnish or vice versa), the average is a preferable metric. Any error value is meaningless when viewed independently; the errors must be compared and contrasted to each other in order to judge the robustness of a relationship or lack thereof.

8 Experiments (Question 2)

8.1 Model

Finding differences in case representation across languages requires a subset of languages and a multilingual model. We used a pre-trained multilingual BERT, downloaded from Hugging Face (Devlin et al., 2018). This model was trained on Wikipedia entries in 104 languages, with the languages being those composed by "largest Wikipedias" (Devlin et al., 2018). The model was initially trained for masked language modeling (MLM). It is also case sensitive, making distinctions between words such as "Linguistics" and "linguistics."

8.2 Languages

German, Finnish, Russian, Polish, and Czech were used for this study. The grammatical cases used to test the "concept of linguistic case across languages" were the accusative, locative, dative, and

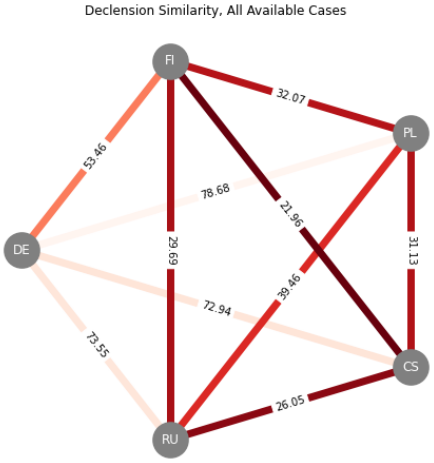
genitive. Any two of these languages have at least two of the selected grammatical cases in common, motivating their use. In addition, the availability of comprehensive corpora of the most common words in all declensions led to these five being used. The accusative case modifies the direct object of a sentence (Drout), such as "linguistics" in "I love linguistics." Similarly, the dative case modifies the indirect object of a sentence (Drout), such as "her" in "I gave [to] her a kiss." The locative case is used in reference to the location of an object (Drout). In "I am in the bedroom," "[the] bedroom" would be declined according to the locative case in a language like Czech. Finnish contains six specific locative cases; instead of choosing one of these six, none were included for the following analyses. While the locative case does exist in Russian, it is often absorbed into the more conventional prepositional case (Leed et al., 1981), and was thus not used for this segment of the study . Finally, the genitive case resembles the possessive case in English (Drout), in the form of adding "apostrophe s" to the end of nouns to denote ownership.

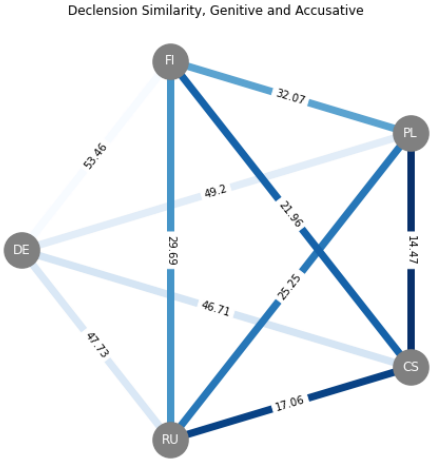
8.3 Data

Analyses were run using web-scraped corpora of the language's most common nouns in the nominative (uncased) form, as well as declensions of that noun in each possible case. Table 6 displays how many unique nouns were used for each of the five languages. The most common nouns were those that occurred most frequently in movie subtitles scraped from Open Subtitles. These were individually run through online dictionaries (Obcych, 1991) or, in the case of German, inner joined with a comprehensive Wikimedia dump of all German nouns and their cases. All the nouns used were singular. For every noun in the corpus, there was a valid declension of each case fed through the model where available for that language. The quantitative, semantic representations of each noun were derived using the average of the Multilingual BERT's sequence of last hidden states. Henceforth, such vectors will be referred to as embeddings.

9 Results (Question 2)

In this section, "loss" refers to the value of the above equation for two different languages, and higher "similarity" corresponds to lower loss. Using pairs of five different languages, ten distinct relative relationships are computable. Figure 1 references the "red" and "blue" graphs formed by plotting the losses between languages in the form of a weighted K_5 graph. For the "blue" graph, the losses from each relationship's mapping are displayed in Figure 1. In both graphs, darker lines correspond to greater similarity and lower loss, while lighter lines imply the converse.





	FI	CS	PL	DE	RU
FI	-	21.96	32.07	53.46	29.69
CS	-	-	14.47	46.71	17.06
PL	-	-	-	49.20	25.25
DE	-	-	-	-	47.73
RU	-	-	-	-	-

Figure 1: Loss results for the linear map with all available cases for the language pair (the top, "red" graph), and loss results for the genitive and accusative linear map (the bottom, "blue" graph). In both images, a darker line indicates a more less error, and a more robust average mapping. Below both is the "blue" graph in the form of a table. The top three losses leading to greatest similarity between two languages are highlighted in green; the lowest, red

The "red" graph shows the losses between the different languages when all cases are used. This

leads to some interesting observations, such as Finnish and Czech being the closest languages to each other, and Polish and German being the farthest. While there is more to unpack here, the comparison is unbalanced in that some languages have more cases in common and more points to map between (Polish and Czech have four), whereas some languages only have two (Finnish and everything else). To balance this, we repeat the least squares computation only including the genitive and accusative cases for all pairs, in order to assess the accuracy of the map without the deductive consequences of fitting to additional cases.

The "blue" graph depicts these similarities and losses when the genitive and accusative case are the only cases used. Because we are fitting fewer parameters to the data, all loss values are less than or equal to their corresponding losses in the "red" graph. The differences in the Polish-Czech connection and Czech-German connection are particularly interesting, as both witness the largest drop in loss by 53.5 percent and 36.0 percent respectively. However, a key takeaway is the list of most similar language pairs:

1. Polish and Czech
2. Russian and Czech
3. Finnish and Czech

Conversely, the most different language pairs are

1. Finnish and German
2. Polish and German
3. Russian and German

This result is to be expected, as similarity correlates with linguistic similarity by family group. German is a Germanic language, and Finnish is a Uralic language. Conversely, the other three are Slavic languages. Two of the three most similar linguistic connections occur between Slavic languages. Overall, there seems to be a rather robust K_3 subgraph defining the similarity between Slavic connections. We explore this more in the discussion.

10 Discussion (Question 2)

These accuracy metrics are reflective of familial similarities in declension. First, one can interpret these results in the context of noun gender. Polish, Czech, and Russian all identify a male, female, and neuter gender (Leed et al., 1981; Obcych, 1991; of the Czech Language, 2008). All three of these languages go one step further, identifying two different declension rules for animate and inanimate nouns, depending upon the case (Leed et al., 1981; Obcych, 1991; of the Czech Language, 2008). Because of this similarity across languages, it is possible that the representation of these features in 768-dimensional space allowed for some form of consistency between the languages, thus facilitating more direct linear maps.

Conversely, Finnish identifies neither gender, nor animacy (Korpela, 1997). Additionally, forming the accusative case necessitates adding "n" or nothing to the end of the noun in the nominative case (Korpela, 1997). From the 2050 word corpus used for testing, every accusative noun shared the exact same spelling of the corresponding nominative noun. Out of these, 943 instances (0.46) of the genitive case were just the accusative case with a "n" added to the end; many others contained some slight variation of the vowels but still ended with "n." For the model to distinguish between Finnish words in different cases, that task would be quite rudimentary, but still distinct from the Slavic rules.

German is a little more difficult. While German contains declension, declensions are more common among articles and demonstrative pronouns (PONS). For nouns, the accusative case can end with "n," but accusative nouns often were equivalent to genitive nouns (in 1051 of 2369 instances, 0.44) in our corpus. Thus, so the model would be able to distinguish between cases for half of our words, we added the definite article corresponding to the case and gender before the word. This added another layer of clarity, and another way to make distinctions. Although articles were added, there is still much room for error because of the articles' ambiguity. For instance, "die," the direct article used for feminine nouns in the accusative case, is also used for nominative feminine nouns (PONS). This casewise use of the article holds for plural nouns in all cases (PONS). Neuter nouns take "das" as the direct article in both the accusative and nominative cases (PONS). Of the 457 neuter nouns in our corpus, the accusative and nominative nouns were equivalent in 100 percent of instances. Thus, without the context of a surrounding sentence, the original model would have no way to distinguish between the accusative and nominative cases. This

ambivalence is simply not present in the Slavic words used to construct the average embedding.

11 Additional Experiments

Two additional experiments were performed following the compelling conclusions to our second question.

11.1 Principal Component Analysis

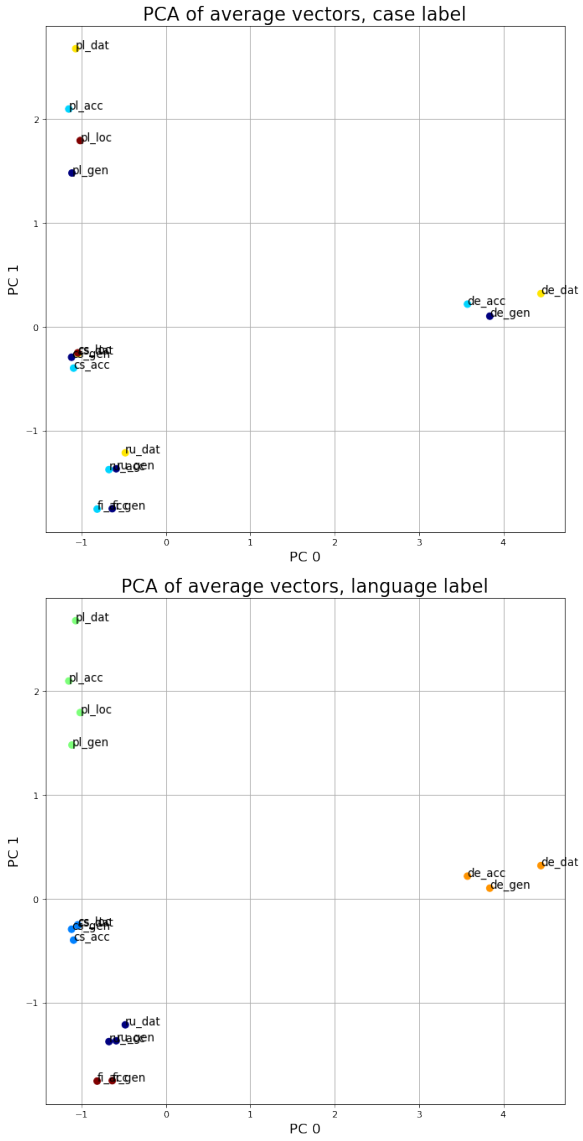


Figure 2: Average word embedding vectors represented in $\mathbb{R}^2$ by the first two principal components. The first graph colors points by case; the second, by language.

In this section, we perform a principal component analysis (PCA) on each mean vector, identifying the directions of maximum variance in the 768-dimensional subspace. Figure 2 shows the 2D plot of the first principal component and the second principal component. There is a clear clustering by language, and not by case. Provided that the vectors were computed as a mean of cases by language, that the first principal components are very indicative of language is not surprising. These constitute 0.438 and 0.211 of the explained variance, respectively.

Our subsequent analyses rely on k-means. K-means is an unsupervised learning algorithm designed to find clusters that optimize squared Euclidean distance between points and cluster centers. By varying the two principal components used, we use k-means to find clusters in $\mathbb{R}^2$ that best align with the cases and the languages, respectively. For the first thirteen principal components, we compute principal components i, j of each vector and apply k-means to the corresponding matrix in $\mathbb{R}^{h\times 2}$. For each cluster, we use accuracy to determine how well this combination of components fits the intended clustering. We find that PC's 3 and 4 (indices 2 and 3) lead to the optimal clustering of language, while 6 and 9 (indices 5 and 8) nicely cluster by case. Figure 3 shows these resulting graphs in $\mathbb{R}^2$.

In addition to this graphic, we find an interesting, distinct tradeoff for the principal component combination that most accurately clusters by language versus by case. This stark shift occurs as one shifts from the 4th component with any other component to the 5th component with any other component. In the former case, language is better classified; the latter, case.

These analyses imply that language creates the most variation in the vectors' values, but after a certain point the variance aligns more with the morphological case. This confirms our expectations as the vectors are fundamentally averages of embedding vectors in different languages. This tradeoff illustrates how case is also encoded in the average vectors, but not as much as language. All principal components up to the fourth explain 87.8 percent of the variance.

11.2 Bidirectional Graph

In the above experiment, we take the average of linear transformation accuracies between two languages, as we seek to compute the overall loss and similarity. However, it is notable that as aforementioned, the losses from mapping x to y and y to x can be different. It may, for instance, be easier for the Russian model to "downgrade" its complex morphological cases when mapping to German,

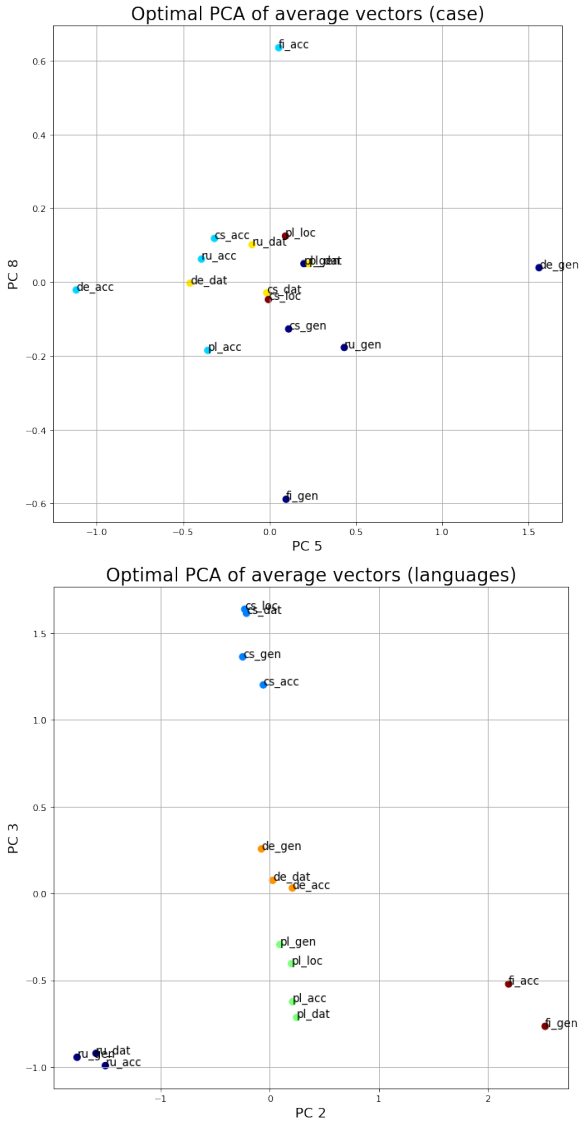


Figure 3: Average word embedding vectors represented in $\mathbb{R}^2$ by the principal components that optimize accuracy for case and language respectively, per k-means.

but the converse does not necessarily hold. Thus, we define the following loss function to compute losses:

$$\ell = ||Y - XA||^2$$

Figure 4 graph shows the results of mapping from the language along the vertical axis to that along the horizontal.

The results from this experiment confirm those above. When mapping from German to any other language, the loss is notably higher every time. The loss is less when mapping to and from Finnish; nonetheless, loss is always the least among the Slavic languages' maps to each other. As a case study, the greater ease of Polish to map to Russian than Russian to Polish is likely caused by the uniqueness of Polish letters such as "ł" and "e," whereas other Cyrillic languages orthographically similar to Russian appear in the corpus (e.g. Ukrainian). This may create ambiguity with regards to the true language of word embeddings, and especially with the true language of common word ending tokens.

Figure 4: Average word embedding vectors represented in $\mathbb{R}^2$ by the principal components that optimize accuracy for case and language respectively, per k-means.

12 Future work

12.1 Question 1

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.

12.2 Question 2

For the second question, it would be interesting to repeat the same process using more languages. There is a sparsity of clean, reliable data containing all cased variants of nouns in a specific language. The five languages tested were largely used due to corpora being available. However, expanding the multilingual task task to languages such as Serbian, Ukrainian, Turkish, Arabic (Modern Standard), and Hungarian may yield interesting conclusions. Furthermore, such languages would allow there to be stronger conclusions regarding the similarities between morphological case representation within and across various language families.

If this experiment were to be repeated as described above, it would also be valuable to have the computing power and comprehensive datasets of all labeled languages to train a new model on just the languages in question. There do exist models such as Slavic BERT (Arkhipov et al., 2019) that may be advantageous in this regard. Since Slavic BERT is trained on every Slavic language we used, we considered using this fine-tuned model for better analyses. However, our conclusions would have been confounded by the lack of familial diversity. Also, as to not only have three languages to produce conclusive results, a corpus of all Bulgarian noun declensions would have had to be found, and that would have been intensive for the obscure language.

Using our multilingual BERT, one can be relatively confident that any word passed into the model exists for a language; however, there is no guarantee that the word is unique to that language. The word "die," for instance, could refer to the feminine definite article in German, "that [one]" in Dutch, or a term relating to loss of life in English. Likewise, specific model tokens could also be shared by many languages, and thus may not be desirable tokenizations for words in some languages while favoring others.

13 Acknowledgements

Thank you Professor Frank and Sophie for a great year and a very well-taught, engaging, and manageable class!

14 Code Availability

Our first main question, regarding the extent to which language models' word embeddings reflected relationships between case within Czech, used code which can be found here.

Our second main question investigated the extent to which language models can agree upon some notion of morphological case between languages. The repository containing the code used resides here.

References

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

Mikhail Arkhipov, Maria Trofimova, Yuri Kuratov, and Alexey Sorokin. 2019. Tuning multilingual transformers for language-specific named entity recognition. In *Proceedings of the 7th Workshop on Balto-Slavic Natural Language Processing*, pages 89–93, Florence, Italy. Association for Computational Linguistics.

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.

Institute of the Czech Language. 2008. Internetová jazyková příručka (internet language guide).

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Michael Drout. Word order and cases.

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).

Jukka Korpela. 1997. Cases in finnish.

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).

Richard Leed, Alexander Nakhimovsky, and Alice Nakhimovsky. 1981. Beginning russian grammar.

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

Szkoła Jezyków Obcych. 1991. Polish online dictionary.

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

PONS. Pons online dictionary.

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

Language	**Words**	**Accusative**	**Locative**	**Dative**	**Genitive**
German (DE)	2369	present	-	present	present
Czech (CS)	1574	present	present	present	present
Polish (PL)	815	present	present	present	present
Russian (RU)	1799	present	-	present	present
Finnish (FI)	2050	present	-	-	present

Table 6: Number of words in each language's corpus used to compute the average, as well as cases among the four selected present in each language.