

# Compound Splitting and Analysis for Russian

Daniil Vodolazsky<sup>1,2</sup>, Hermann Petrov<sup>1</sup>

<sup>1</sup>Sber

<sup>2</sup>Higher School of Economics

September 10th, 2021

# DeriMo 2021

## ① Introduction

## ② Compounding in Russian

## ③ Models

- Compound Identification
- Splitting and Analysis
- Hypotheses Scoring

## ④ Training, Evaluation, and Error Analysis

- Compound Identification
- Hypotheses Generation and Scoring

## ⑤ Conclusion and Future Work

# Table of Contents

- 1 Introduction
- 2 Compounding in Russian
- 3 Models
  - Compound Identification
  - Splitting and Analysis
  - Hypotheses Scoring
- 4 Training, Evaluation, and Error Analysis
  - Compound Identification
  - Hypotheses Generation and Scoring
- 5 Conclusion and Future Work

- A **compound** is a lexeme that consists of two or more stems.

- A **compound** is a lexeme that consists of two or more stems.
- **Compounding** is a word-formation process that creates such lexemes.

- A **compound** is a lexeme that consists of two or more stems.
- **Compounding** is a word-formation process that creates such lexemes.
- It can be highly productive in synthetic languages.

- A **compound** is a lexeme that consists of two or more stems.
- **Compounding** is a word-formation process that creates such lexemes.
- It can be highly productive in synthetic languages.
- German: many splitters exist, the GermaNet dataset.

- A **compound** is a lexeme that consists of two or more stems.
- **Compounding** is a word-formation process that creates such lexemes.
- It can be highly productive in synthetic languages.
- German: many splitters exist, the GermaNet dataset.
- Russian: no open-source tools or resources.

- A **compound** is a lexeme that consists of two or more stems.
- **Compounding** is a word-formation process that creates such lexemes.
- It can be highly productive in synthetic languages.
- German: many splitters exist, the GermaNet dataset.
- Russian: no open-source tools or resources.
- According to (Gromenko, 2020), **32%** of the neologisms in Russian are produced with compounding which makes it important to have a tool for their analysis.

- Although the number of productive compounding patterns in Russian is relatively small, there is structural ambiguity in many cases.

- Although the number of productive compounding patterns in Russian is relatively small, there is structural ambiguity in many cases.

железнодорожный 'zhel'eznodorozhnyy' (*railway-related*) matches the following three rules:

- **rule761**([adj + ITFX] + noun + н1(ый) → adj): железный(-ая), дорога 'zhel'eznyy(-aya), doroga' (*railway, lit. iron road*);
- **rule754**([adj + ITFX] + adj → adj): железный, дорожный 'zhel'eznyy, dorozhnyy';
- **rule776**(adv + adj → adj): железно, дорожный 'zhel'ezno, dorozhnyy'.

- Although the number of productive compounding patterns in Russian is relatively small, there is structural ambiguity in many cases.  
железнодорожный 'zhel'eznodorozhnyy' (*railway-related*) matches the following three rules:
  - **rule761**([adj + ITFX] + noun + н1(ый) → adj): железный(-ая), дорога 'zhel'eznyy(-aya), doroga' (*railway, lit. iron road*);
  - **rule754**([adj + ITFX] + adj → adj): железный, дорожный 'zhel'eznyy, dorozhnyy';
  - **rule776**(adv + adj → adj): железно, дорожный 'zhel'ezno, dorozhnyy'.
- The correct choice depends on the semantics of the source and produced words.

- Although the number of productive compounding patterns in Russian is relatively small, there is structural ambiguity in many cases.  
железнодорожный 'zhel'eznodorozhnyy' (*railway-related*) matches the following three rules:
  - **rule761**([adj + ITFX] + noun + н1(ый) → adj): железный(-ая), дорога 'zhel'eznyy(-aya), doroga' (*railway, lit. iron road*);
  - **rule754**([adj + ITFX] + adj → adj): железный, дорожный 'zhel'eznyy, dorozhnyy';
  - **rule776**(adv + adj → adj): железно, дорожный 'zhel'ezno, dorozhnyy'.
- The correct choice depends on the semantics of the source and produced words.
- A morphemic segmentation in all three analysis would be the same: *zhel'ez|n|o|dorozh|n|yy*—and the structural information would be lost.

- Although the number of productive compounding patterns in Russian is relatively small, there is structural ambiguity in many cases.  
железнодорожный 'zhel'eznodorozhnyy' (*railway-related*) matches the following three rules:
  - **rule761**([adj + ITFX] + noun + н1(ый) → adj): железный(-ая), дорога 'zhel'eznyy(-aya), doroga' (*railway, lit. iron road*);
  - **rule754**([adj + ITFX] + adj → adj): железный, дорожный 'zhel'eznyy, dorozhnyy';
  - **rule776**(adv + adj → adj): железно, дорожный 'zhel'ezno, dorozhnyy'.
- The correct choice depends on the semantics of the source and produced words.
- A morphemic segmentation in all three analysis would be the same: *zhel'ez|n|o|dorozh|n|yy*—and the structural information would be lost.
- Therefore it is desirable to provide the analysis that includes the lemmas of the source words and (optionally) the rule ID.

# Table of Contents

- 1 Introduction
- 2 Compounding in Russian
- 3 Models
  - Compound Identification
  - Splitting and Analysis
  - Hypotheses Scoring
- 4 Training, Evaluation, and Error Analysis
  - Compound Identification
  - Hypotheses Generation and Scoring
- 5 Conclusion and Future Work

- **Pure compounds**  $w_1 + w_2$ :
  - северо-запад 'severo-zapad' (*northwest*);
  - нефте|промышленность 'nefte|promyshlennost'' (*oil industry*);
  - цельно|металлический 'tsel'no|metallicheskiy' (*full metal*).

- **Pure compounds**  $w_1 + w_2$ :
  - северо-запад '*severo-zapad*' (*northwest*);
  - нефте|промышленность '*nefte|promyshlennost*' (*oil industry*);
  - цельно|металлический '*tsel'no|metallicheskij*' (*full metal*).
- Compounds of the same structure, without a linking element:
  - рыба-меч '*ryba-mech*' (*swordfish*, *lit. fish-sword*);
  - коронавирус '*koronavirus*' (*coronavirus*).

- **Pure compounds**  $w_1 + w_2$ :
  - северо-запад 'severo-zapad' (*northwest*);
  - нефте|промышленность 'nefte|promyshlennost'' (*oil industry*);
  - цельно|металлический 'tsel'no|metallicheskiy' (*full metal*).
- Compounds of the same structure, without a linking element:
  - рыба-меч 'ryba-mech' (*swordfish, lit. fish-sword*);
  - коронавирус 'koronavirus' (*coronavirus*).
- **Parasyntetic compounds**, most commonly  $w_1 + w_2 + sfx$ :
  - зелено|глазый 'zeleno|glaz|yy' (*green-eyed*), but not \*зеленоглаз(а) 'zelenoglaz(a)', \*глазый 'glazyi'.

- **Pure compounds**  $w_1 + w_2$ :
  - северо-запад 'severo-zapad' (*northwest*);
  - нефте|промышленность 'nefte|promyshlennost'' (*oil industry*);
  - цельно|металлический 'tsel'no|metallicheskiy' (*full metal*).
- Compounds of the same structure, without a linking element:
  - рыба-меч 'ryba-mech' (*swordfish, lit. fish-sword*);
  - коронавирус 'koronavirus' (*coronavirus*).
- **Parasyntetic compounds**, most commonly  $w_1 + w_2 + sfx$ :
  - зелено|глазый 'zeleno|glaz/yu' (*green-eyed*), but not \*зеленоглаз(а) 'zelenoglaz(a)', \*глазый 'glazyi'.
- **Classical and neoclassical compounds**  $r_1 + r_2 (+sfx)$ :
  - био|лог|ия 'bio|log|iya' (*biology*);
  - психо|пат 'psikho|pat' (*psychopath*).

- **Pure compounds**  $w_1 + w_2$ :
  - северо-запад 'severo-zapad' (*northwest*);
  - нефте|промышленность 'nefte|promyshlennost'' (*oil industry*);
  - целно|металлический 'tsel'no|metallicheskiy' (*full metal*).
- Compounds of the same structure, without a linking element:
  - рыба-меч 'ryba-mech' (*swordfish, lit. fish-sword*);
  - коронавирус 'koronavirus' (*coronavirus*).
- **Parasyntetic compounds**, most commonly  $w_1 + w_2 + sfx$ :
  - зелено|глазый 'zeleno|glaz/yu' (*green-eyed*), but not \*зеленоглаз(а) 'zelenoglaz(a)', \*глазый 'glazyi'.
- **Classical and neoclassical compounds**  $r_1 + r_2 (+sfx)$ :
  - био|лог|ия 'bio|log|iya' (*biology*);
  - психо|пат 'psikho|pat' (*psychopath*).
- Such roots and other international morphemes can act as quasi-affixes and attach to normal Russian stems:
  - игро|тека 'igro|teka' (*playroom*), cf. библио|тека 'biblio|teka' (*library*);
  - мега|завод 'mega|zavod' (*megafactory*).

- **Pure compounds**  $w_1 + w_2$ :
  - северо-запад 'severo-zapad' (*northwest*);
  - нефте|промышленность 'nefte|promyshlennost'' (*oil industry*);
  - цельно|металлический 'tsel'no|metallicheskii' (*full metal*).
- Compounds of the same structure, without a linking element:
  - рыба-меч 'ryba-mech' (*swordfish, lit. fish-sword*);
  - коронавирус 'koronavirus' (*coronavirus*).
- **Parasyntetic compounds**, most commonly  $w_1 + w_2 + sfx$ :
  - зелено|глазый 'zeleno|glaz|yy' (*green-eyed*), but not \*зеленоглаз(а) 'zelenoglaz(a)', \*глазый 'glazyi'.
- **Classical and neoclassical compounds**  $r_1 + r_2 (+sfx)$ :
  - био|лог|ия 'bio|log|iya' (*biology*);
  - психо|пат 'psikho|pat' (*psychopath*).
- Such roots and other international morphemes can act as quasi-affixes and attach to normal Russian stems:
  - игро|тека 'igro|teka' (*playroom*), cf. библио|тека 'biblio|teka' (*library*);
  - мега|завод 'mega|zavod' (*megafactory*).
- **Phrasal compounds** (sometimes with affixation):
  - выше|упомянутый 'vyshe|upomyanutyy' (*above-mentioned*);
  - с|ума|сшедший 's|uma|sshedshiy' (*mad, crazy, lit. gone out of mind*);
  - с|ума|сшествие 's|uma|sshestvie' (*madness*), but not \*сшествие 'sshestvie';
  - ничего|не|делание 'nichego|ne|delanie' (*doing nothing*).

# Table of Contents

- 1 Introduction
- 2 Compounding in Russian
- 3 **Models**
  - Compound Identification
  - Splitting and Analysis
  - Hypotheses Scoring
- 4 Training, Evaluation, and Error Analysis
  - Compound Identification
  - Hypotheses Generation and Scoring
- 5 Conclusion and Future Work

We separate our pipeline into three stages:

We separate our pipeline into three stages:

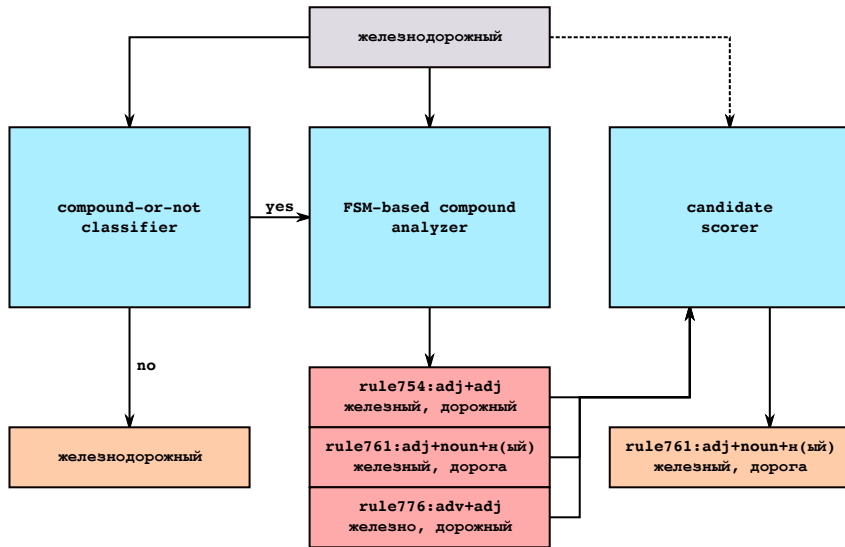
- 1 **Compound identification:** classify whether a word is a compound.

We separate our pipeline into three stages:

- 1 **Compound identification**: classify whether a word is a compound.
- 2 **Splitting and Analysis**: generation of the possible analyses for the compound.

We separate our pipeline into three stages:

- 1 **Compound identification**: classify whether a word is a compound.
- 2 **Splitting and Analysis**: generation of the possible analyses for the compound.
- 3 **Hypotheses Scoring**: score the candidates and select the best.



**Figure:** Diagram of the proposed pipeline. Gray: input (*zhel'eznodorozhnyy*), blue: algorithmic blocks, red: analyses generated by a rule-based model, orange: output (final analyses). The dotted line indicates that the target word is not always used in a scoring process.

# Compound Identification

First, we classify whether a word is a compound. We solve this task as a binary classification problem with a character-level bidirectional LSTM with attention. More formally, given a sequence of input character embeddings (including the special characters for the beginning and ending of the word)  $\mathbf{x}_1, \dots, \mathbf{x}_n$ , we compute

# Compound Identification

First, we classify whether a word is a compound. We solve this task as a binary classification problem with a character-level bidirectional LSTM with attention. More formally, given a sequence of input character embeddings (including the special characters for the beginning and ending of the word)  $\mathbf{x}_1, \dots, \mathbf{x}_n$ , we compute

$$\mathbf{h}_1^F, \dots, \mathbf{h}_n^F, \mathbf{c}^F = \text{LSTM}^F(\mathbf{x}_1, \dots, \mathbf{x}_n);$$

$$\mathbf{h}_1^B, \dots, \mathbf{h}_n^B, \mathbf{c}^B = \text{LSTM}^B(\mathbf{x}_1, \dots, \mathbf{x}_n);$$

$$\mathbf{h}_t = [\mathbf{h}_t^F; \mathbf{h}_t^B], t = 1, \dots, n;$$

$$\mathbf{q} = \mathbf{Q}[\mathbf{c}^F; \mathbf{c}^B];$$

$$\mathbf{a} = \text{MultiheadSelfAttention}(\mathbf{q}, (\mathbf{h}_1, \dots, \mathbf{h}_n));$$

$$\mathbf{y} = \sigma(\mathbf{W}\mathbf{a}).$$

# Compound Identification

First, we classify whether a word is a compound. We solve this task as a binary classification problem with a character-level bidirectional LSTM with attention. More formally, given a sequence of input character embeddings (including the special characters for the beginning and ending of the word)  $\mathbf{x}_1, \dots, \mathbf{x}_n$ , we compute

$$\begin{aligned}\mathbf{h}_1^F, \dots, \mathbf{h}_n^F, \mathbf{c}^F &= \text{LSTM}^F(\mathbf{x}_1, \dots, \mathbf{x}_n); \\ \mathbf{h}_1^B, \dots, \mathbf{h}_n^B, \mathbf{c}^B &= \text{LSTM}^B(\mathbf{x}_1, \dots, \mathbf{x}_n); \\ \mathbf{h}_t &= [\mathbf{h}_t^F; \mathbf{h}_t^B], t = 1, \dots, n; \\ \mathbf{q} &= \mathbf{Q}[\mathbf{c}^F; \mathbf{c}^B]; \\ \mathbf{a} &= \text{MultiheadSelfAttention}(\mathbf{q}, (\mathbf{h}_1, \dots, \mathbf{h}_n)); \\ \mathbf{y} &= \sigma(\mathbf{W}\mathbf{a}).\end{aligned}$$

We used the A. N. Tikhonov dictionary with morpheme segmentation as the dataset:

# Compound Identification

First, we classify whether a word is a compound. We solve this task as a binary classification problem with a character-level bidirectional LSTM with attention. More formally, given a sequence of input character embeddings (including the special characters for the beginning and ending of the word)  $\mathbf{x}_1, \dots, \mathbf{x}_n$ , we compute

$$\begin{aligned}\mathbf{h}_1^F, \dots, \mathbf{h}_n^F, \mathbf{c}^F &= \text{LSTM}^F(\mathbf{x}_1, \dots, \mathbf{x}_n); \\ \mathbf{h}_1^B, \dots, \mathbf{h}_n^B, \mathbf{c}^B &= \text{LSTM}^B(\mathbf{x}_1, \dots, \mathbf{x}_n); \\ \mathbf{h}_t &= [\mathbf{h}_t^F; \mathbf{h}_t^B], t = 1, \dots, n; \\ \mathbf{q} &= \mathbf{Q}[\mathbf{c}^F; \mathbf{c}^B]; \\ \mathbf{a} &= \text{MultiheadSelfAttention}(\mathbf{q}, (\mathbf{h}_1, \dots, \mathbf{h}_n)); \\ \mathbf{y} &= \sigma(\mathbf{W}\mathbf{a}).\end{aligned}$$

We used the A. N. Tikhonov dictionary with morpheme segmentation as the dataset:

- громкоголосый громк:**ROOT**/о:LINK/голос:**ROOT**/ый:END/ый:END → compound;

# Compound Identification

First, we classify whether a word is a compound. We solve this task as a binary classification problem with a character-level bidirectional LSTM with attention. More formally, given a sequence of input character embeddings (including the special characters for the beginning and ending of the word)  $\mathbf{x}_1, \dots, \mathbf{x}_n$ , we compute

$$\begin{aligned} \mathbf{h}_1^F, \dots, \mathbf{h}_n^F, \mathbf{c}^F &= \text{LSTM}^F(\mathbf{x}_1, \dots, \mathbf{x}_n); \\ \mathbf{h}_1^B, \dots, \mathbf{h}_n^B, \mathbf{c}^B &= \text{LSTM}^B(\mathbf{x}_1, \dots, \mathbf{x}_n); \\ \mathbf{h}_t &= [\mathbf{h}_t^F; \mathbf{h}_t^B], t = 1, \dots, n; \\ \mathbf{q} &= \mathbf{Q}[\mathbf{c}^F; \mathbf{c}^B]; \\ \mathbf{a} &= \text{MultiheadSelfAttention}(\mathbf{q}, (\mathbf{h}_1, \dots, \mathbf{h}_n)); \\ \mathbf{y} &= \sigma(\mathbf{W}\mathbf{a}). \end{aligned}$$

We used the A. N. Tikhonov dictionary with morpheme segmentation as the dataset:

- громкоголосый громк:**ROOT**/о:LINK/голос:**ROOT**/ый:END/ый:END → compound;
- врать вр:**ROOT**/а:SUFF/ть:SUFF → not compound.

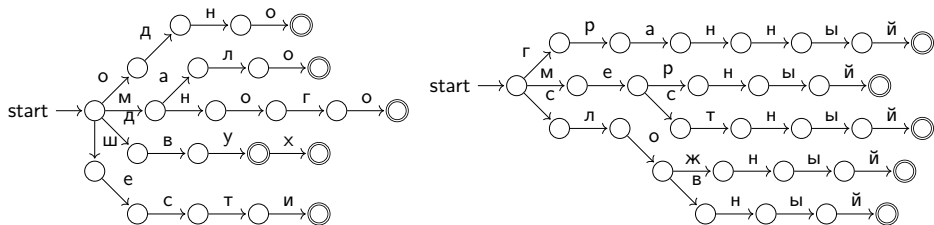
```
{
  "rule_id": "rule761([num + ITFX] + noun + н1(ый) -> adj)",
  "info": "COMPOUND,SFX",
  "pos_b": "noun",
  "pos_a": "adj",
  "compound_rules": {
    "head": ["rule619*(noun + н1(ый) -> adj)"],
    "modifiers": [{"pos_m": "num", "complex": ["ruleINTERFIX(num)"}]},
    "order": [1, 0],
    "before_merge": [],
    "after_merge": []
  }
}
```

# Splitting and Analysis: FSMs

We use the FSM concatenation technique to find all possible valid compound partitions. Most of the Russian compounds can be described with the regular expression  $l(i)r(s)$ , where  $l$  is the left word,  $r$  is the right word,  $i$  is the interfix applied to the left stem (optional), and  $s$  is a suffix (optional).

# Splitting and Analysis: FSMs

We use the FSM concatenation technique to find all possible valid compound partitions. Most of the Russian compounds can be described with the regular expression  $l(i)r(s)$ , where  $l$  is the left word,  $r$  is the right word,  $i$  is the interfix applied to the left stem (optional), and  $s$  is a suffix (optional).



**Figure:** Left: the FSM for numerals with interfix (один 'odin' (one) → одно 'odno', два 'dva' (two) → два/двух 'dva/dvukh', шесть 'shest'' (six) → шести 'shesti', мало 'malo' (few) and много 'mnogo' (many) remain unchanged). Right: the FSM for adjectives derived from nouns грань 'gran'' (face (of a figure)), мера 'mera' (measure), место 'mesto' (place, seat), слог 'slog' (syllable), слово 'slovo' (word) with the suffix -н1(ый) '-n1(yy)'. The concatenated FSM with ID **rule761**([num + ITFX] + noun + н1(ый) → adj) can recognize words многомерный 'mnogomernyy' (multidimensional), однословный 'odnoslovnyy' (one-word), etc. The corresponding analyses would be represented in the form (**rule761**([num + ITFX] + noun + н1(ый) → adj), один, слово).

# Hypotheses Scoring: Baselines

- 1 Random score from a uniform distribution:  $F \sim \mathcal{U}(0,1)$ .

# Hypotheses Scoring: Baselines

- 1 Random score from a uniform distribution:  $F \sim \mathcal{U}(0, 1)$ .
- 2 Frequency addition on lemmas:  $F(l, r) = \#(l) + \#(r)$ .

# Hypotheses Scoring: Baselines

- 1 Random score from a uniform distribution:  $F \sim \mathcal{U}(0, 1)$ .
- 2 Frequency addition on lemmas:  $F(l, r) = \#(l) + \#(r)$ .
- 3 Frequency multiplication on lemmas:  $F(l, r) = \max(1, \#(l)) \cdot \max(1, \#(r))$ .

# Hypotheses Scoring: Baselines

- 1 Random score from a uniform distribution:  $F \sim \mathcal{U}(0, 1)$ .
- 2 Frequency addition on lemmas:  $F(l, r) = \#(l) + \#(r)$ .
- 3 Frequency multiplication on lemmas:  $F(l, r) = \max(1, \#(l)) \cdot \max(1, \#(r))$ .
- 4 PMI-based score on paradigms:

$$F(l, r) = \log_2 \left( \frac{\max(1, \sum_{l_f \in C_l} \sum_{r_f \in C_r} \#(l_f, r_f) + \#(r_f, l_f))}{\max(1, \sum_{l_f \in C_l} \#(l_f)) \cdot \max(1, \sum_{r_f \in C_r} \#(r_f))} \right).$$

- 1 Random score from a uniform distribution:  $F \sim \mathcal{U}(0, 1)$ .
- 2 Frequency addition on lemmas:  $F(l, r) = \#(l) + \#(r)$ .
- 3 Frequency multiplication on lemmas:  $F(l, r) = \max(1, \#(l)) \cdot \max(1, \#(r))$ .
- 4 PMI-based score on paradigms:

$$F(l, r) = \log_2 \left( \frac{\max(1, \sum_{l_f \in C_l} \sum_{r_f \in C_r} \#(l_f, r_f) + \#(r_f, l_f))}{\max(1, \sum_{l_f \in C_l} \#(l_f)) \cdot \max(1, \sum_{r_f \in C_r} \#(r_f))} \right).$$

- 5 Cosine of two lemmas based on word embeddings:

$$F(l, r) = \cos(\text{emb}(l), \text{emb}(r));$$

- 1 Random score from a uniform distribution:  $F \sim \mathcal{U}(0, 1)$ .
- 2 Frequency addition on lemmas:  $F(l, r) = \#(l) + \#(r)$ .
- 3 Frequency multiplication on lemmas:  $F(l, r) = \max(1, \#(l)) \cdot \max(1, \#(r))$ .
- 4 PMI-based score on paradigms:

$$F(l, r) = \log_2 \left( \frac{\max(1, \sum_{l_f \in C_l} \sum_{r_f \in C_r} \#(l_f, r_f) + \#(r_f, l_f))}{\max(1, \sum_{l_f \in C_l} \#(l_f)) \cdot \max(1, \sum_{r_f \in C_r} \#(r_f))} \right).$$

- 5 Cosine of two lemmas based on word embeddings:

$$F(l, r) = \cos(\text{emb}(l), \text{emb}(r));$$

- 6 Cosine of three lemmas based on word embedding:

$$F(l, r, c) = \cos \left( \text{emb}(c), \frac{\text{emb}(l)}{\|\text{emb}(l)\|} + \frac{\text{emb}(r)}{\|\text{emb}(r)\|} \right).$$

- In contrast to the baseline models, a neural model can fit on training data.
- We designed the model's architecture in a way that allows loading weights pre-trained on another, high-resource task, such as compound-or-not classification.
- **Input:** a compound word  $c$  and  $N$  hypotheses in the form  $(l, r, R_l, R_r, R_c)$ . Additionally, we define three special tokens  $p_l, p_r, p_c$  with the corresponding embeddings to give a model the information about words positions in the analysis.

- In contrast to the baseline models, a neural model can fit on training data.
- We designed the model's architecture in a way that allows loading weights pre-trained on another, high-resource task, such as compound-or-not classification.
- **Input:** a compound word  $c$  and  $N$  hypotheses in the form  $(l, r, R_l, R_r, R_c)$ . Additionally, we define three special tokens  $p_l, p_r, p_c$  with the corresponding embeddings to give a model the information about words positions in the analysis.
- **Model training.** For each hypothesis the model independently processes  $(c, R_c, p_c)$ ,  $(l, R_l, p_l)$ ,  $(r, R_r, p_r)$  through a shared BiLSTM. Next, the attention is applied to the united BiLSTM outputs. A query vector is combined from the last cell states. The result of the attention is a vector that is finally fed into the classification head.

# Table of Contents

- 1 Introduction
- 2 Compounding in Russian
- 3 Models
  - Compound Identification
  - Splitting and Analysis
  - Hypotheses Scoring
- 4 Training, Evaluation, and Error Analysis
  - Compound Identification
  - Hypotheses Generation and Scoring
- 5 Conclusion and Future Work

We trained the model with a binary cross-entropy objective. The model hyperparameters and the training parameters:

Parameter	Value	Hyperparameter	Value
number of epochs	30	embedding size	128
batch size	128	embedding dropout	0.1
optimizer	Adam	body	BiLSTM
learning rate	1e-4	body hidden size	256
scheduler	constant	body dropout	0.25
objective	binary cross-entropy	number of body layers	2

**Table:** Training parameters (left) and model hyperparameters (right) for the compound-or-not classification task.

# Evaluation and Error Analysis. Compound Identification

We used 10% of the training data for validation and got 64831, 7203, 24012 samples for training, validation, and test, respectively. We selected the best checkpoint according to a validation F1 score. The resulting model achieved precision **0.9404**, recall **0.9256** and F1-measure **0.9329** on a test set.

# Evaluation and Error Analysis. Compound Identification

We used 10% of the training data for validation and got 64831, 7203, 24012 samples for training, validation, and test, respectively. We selected the best checkpoint according to a validation F1 score. The resulting model achieved precision **0.9404**, recall **0.9256** and F1-measure **0.9329** on a test set.

Error Type	Examples
incorrect gold annotation hyphened prefixes compound-like beginning compound-like ending	ломанос, невоеннообязанный, автономный по-буднишнему, по-военному, экс-король столыпинщина, семафор задубелый, внеочередной
incorrect gold annotation loanwords phrasal abbreviations	враскачку, ботвинья ватерпольный, ватержакетный, миастения комбикорм, помдиректора, торгпредство

**Table:** Main error types in the compound-or-not classification task. Top: false positives, bottom: false negatives.

# Data. Hypotheses Generation and Scoring

We manually collected 1729 compounds with their gold-standard analyses. All compounds are taken from 'Russkaya Grammatika'. Then we split them into training (1143), validation (160), and test sets (364).

rule_id	compound	left	right
<b>rule579</b> ([noun + ITFX] + verb + 0m2 → noun)	водопад	вода	падать
<b>rule754</b> ([num + ITFX] + adj → adj)	стоцентный	сто	процентный
<b>rule767</b> ([noun + ITFX] + verb + n1(ый) → adj)	травоядный	трава	есть
<b>rule961</b> ([adj + ITFX] + verb → verb)	взаимодействовать	взаимный	действовать

**Table:** Samples from the evaluation dataset: водопад 'vodopad' (*waterfall*), стоцентный 'stoprotsentnyy' (*one-hundred-percent*), травоядный 'travoyadnyy' (*herbivorous*), взаимодействовать 'vzaimodeystvovat'' (*interact*).

$$P = \frac{\#(\text{correct analyses})}{\#(\text{total analyses})}; R = \frac{\#(\text{correct analyses})}{\#(\text{total examples})}; F1 = \frac{2PR}{P + R}.$$

$$P = \frac{\#(\text{correct analyses})}{\#(\text{total analyses})}; R = \frac{\#(\text{correct analyses})}{\#(\text{total examples})}; F1 = \frac{2PR}{P + R}.$$

The model achieved precision **0.0748**, recall **0.7796**, and F1-measure **0.1366** on a test set.

False Negatives:

## False Negatives:

- Samples do not match the pattern  $l(i)r(s)$ :
  - в|пол|голоса 'v|pol|golosa' (*in an undertone*);
  - газо|нефте|хранилище 'gazo|nefte|khranilische' (*oil and gas storage*).

## False Negatives:

- Samples do not match the pattern  $l(i)r(s)$ :
  - в|пол|голоса 'v|pol|golosa' (*in an undertone*);
  - газо|нефте|хранилище 'gazo|nefte|khranilische' (*oil and gas storage*).
- Rare wordforms in one of the compound parts:
  - троеборец 'troeborets' (*triathlete*);
  - славянофил 'slavyanofil' (*slavophile, lit. slavs lover*);
  - метеоусловия 'meteousloviya' (*weather conditions*).

## False Negatives:

- Samples do not match the pattern  $l(i)r(s)$ :
  - в|пол|голоса 'v|pol|golosa' (*in an undertone*);
  - газо|нефте|хранилище 'gazo|nefte|khranilische' (*oil and gas storage*).
- Rare wordforms in one of the compound parts:
  - троеборец 'troeborets' (*triathlete*);
  - славянофил 'slavyanofil' (*slavophile, lit. slavs lover*);
  - метеоусловия 'meteousloviya' (*weather conditions*).
- Stems overlapping:
  - чеховед 'chekhoved' (*an expert in Chekhov's literary works*).

## False Negatives:

- Samples do not match the pattern  $l(i)r(s)$ :
  - в|пол|голоса 'v|pol|golosa' (*in an undertone*);
  - газо|нефте|хранилище 'gazo|nefte|khranilische' (*oil and gas storage*).
- Rare wordforms in one of the compound parts:
  - троеборец 'troeborets' (*triathlete*);
  - славянофил 'slavyanofil' (*slavophile, lit. slavs lover*);
  - метеоусловия 'meteousloviya' (*weather conditions*).
- Stems overlapping:
  - чеховед 'chekhoved' (*an expert in Chekhov's literary works*).
- Mistakes in the DerivBase.Ru rules
  - желтощёк 'zheltoschyok' (*yellowcheek*);
  - шелкопряд 'shelkopryad' (*silkworm, lit. silk spinner*).

## False Negatives:

- Samples do not match the pattern  $l(i)r(s)$ :
  - $v|пол|голоса$  'v|pol|golosa' (in an undertone);
  - $газо|нефте|хранилище$  'gazo|nefte|khranilishe' (oil and gas storage).
- Rare wordforms in one of the compound parts:
  - $троеборец$  'troeborets' (triathlete);
  - $славянофил$  'slavyanofil' (slavophile, lit. slavs lover);
  - $метеоусловия$  'meteousloviya' (weather conditions).
- Stems overlapping:
  - $чеховед$  'chekhoved' (an expert in Chekhov's literary works).
- Mistakes in the DerivBase.Ru rules
  - $желтощёк$  'zheltoschyok' (yellowcheek);
  - $шелкопряд$  'shelkopryad' (silkworm, lit. silk spinner).
- Words that do not belong to any productive word-formation pattern:
  - $боеготовный$  'boegotovnyy' (combat-ready);
  - $первогодок$  'pervogodok' (first-year).

## False Negatives:

- Samples do not match the pattern  $l(i)r(s)$ :
  - в|пол|голоса 'v|pol|golosa' (in an undertone);
  - газо|нефте|хранилище 'gazo|nefte|khranilishe' (oil and gas storage).
- Rare wordforms in one of the compound parts:
  - троеборец 'troeborets' (triathlete);
  - славянофил 'slavyanofil' (slavophile, lit. slavs lover);
  - метеоусловия 'meteousloviya' (weather conditions).
- Stems overlapping:
  - чеховед 'chekhoved' (an expert in Chekhov's literary works).
- Mistakes in the DerivBase.Ru rules
  - желтощёк 'zheltoschyok' (yellowcheek);
  - шелкопряд 'shelkopryad' (silkworm, lit. silk spinner).
- Words that do not belong to any productive word-formation pattern:
  - боеготовный 'boegotovnyy' (combat-ready);
  - первогодок 'pervogodok' (first-year).
- Phrasal compounds different from (adv + adj/part → adj):
  - шапкозакидательский 'shapkozakidatel'skiy' (cocksure, lit. related to throwing caps).

# Evaluation. Hypotheses Scoring

To evaluate scoring models, we use the accuracy metric, i. e. the percentage of matches of top-1 best candidates and ground-truth analyses.

Model	Accuracy
Random (100 runs)	24.89 $\pm$ 1.62
Freq. additive (lemmas)	41.05
Freq. multiplicative (lemmas)	42.70
PMI-based (paradigms)	22.59
Cosine, two lemmas	42.42
Cosine, three lemmas	27.54
neural net, trained from scratch (30 epochs, batch size 8)	57.30
neural net, pretrained, zero-shot	31.96
neural net, pretrained, fine-tuned (30 epochs, batch size 8)	<b>60.33</b>

Table: Results of the scoring models.

# Table of Contents

## ① Introduction

## ② Compounding in Russian

## ③ Models

Compound Identification

Splitting and Analysis

Hypotheses Scoring

## ④ Training, Evaluation, and Error Analysis

Compound Identification

Hypotheses Generation and Scoring

## ⑤ Conclusion and Future Work

- We proposed a pipeline for compound identification, splitting and analysis for Russian.
- We collected and annotated a gold standard dataset for Russian compound analysis.
- We achieve a high F1 score on a compound classification task and a high recall score on a hypothesis generation task.
- We compared the different scoring functions based on word frequencies, distributional semantics and neural networks.

- Design and train an end-to-end pipeline.
- Replace LSTMs with convolutional neural networks or transformers.
- Apply the algorithm to a large-scale vocabulary and integrate compounds into the Russian part of Universal Derivations (UDer).
- Collect a larger dataset for training and evaluation of compound analysis.