

Compound Splitting and Analysis for Russian

Daniil Vodolazsky
Sber; Higher School of Economics
Moscow, Russia
daniil.vodolazsky@mail.ru

Hermann Petrov
Sber
Moscow, Russia
gerpetrum@yandex.ru

Abstract

This paper presents a method for compound identification, splitting, and analysis for Russian. First, we identify whether a word is a compound with a neural network. Then we apply a rule-based approach to generate a set of linguistically possible hypotheses with analyses, including the normalization of the compound components. Finally, we score and rank the hypotheses using three techniques: word frequencies, word embeddings, neural networks. We evaluate models on a manually collected and annotated test dataset. We make the dataset and code publicly available.

1 Introduction

Due to the productivity of most languages, it is possible to generate an infinite number of different words, and the speakers can freely form and analyze them even if they have not seen them before. However, this becomes a challenge for computational models in various NLP tasks. Word-level models suffer from a lack of understanding of an internal word structure and cannot process unseen tokens.

The most common approach nowadays is the byte-pair encoding (BPE) (Sennrich et al., 2015), but its segmentation strategy significantly depends on a training corpora domain. All these methods are data-driven and language-independent. It has been shown (Sennrich and Haddow, 2015; Li et al., 2018; Hofmann et al., 2021) that prior knowledge about the language improves the models' performance.

The goal of this work is to propose a linguistically grounded approach to subword segmentation for Russian compounds.

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 such as German or Russian. Many compound splitting and analyzing tools exist for German (Koehn and Knight, 2003; Schmid et al., 2004; Henrich and Hinrichs, 2011; Weller-Di Marco, 2017), and, to the best of our knowledge, no for Russian. 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.

This task is complicated for several reasons. Although the number of productive compounding patterns in Russian is relatively small (compared to the derivational ones), there is structural ambiguity in many cases. For instance, the adjective 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. As shown in the analyses above, the endings of the left items are removed and replaced with the *-o/-e-* interfix¹; the stem of the word

¹The choice of a vowel depends on a previous consonant, e. g. *-e-* is used after palatalized consonants.

дорога ‘*doroga*’ (*road*) is changed with a velar—sibilant alternation before the derivational suffix -н1(ый) ‘-n1(yy)’. 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 not just the splitting but also the analysis that includes the lemmas of the source words and (optionally) the rule ID.

In this paper, we propose an algorithm for splitting and analysis of Russian compounds. Our method is a combination of rules, finite state machines, and deep learning. We manually collected a set of 1.7K ground-truth compound analyses and used it to compare the performance of the models. We make our code and data freely available on GitHub².

2 Compounding in Russian

In this section, we overview the principal categories of Russian compounds. For each of them, we provide a brief description and give examples. Note that our categorization is not Russian-specific and can be applied to other languages, such as Polish (Szymanek, 2017).

Compounding is a word-formation process of concatenating two or more stems. It is often accompanied by morphophonological alternations in a place of the concatenation of stems. The results of compounding are **compounds**. In this paper, to avoid terminological ambiguity, we call them **pure compounds**. Examples of pure compounds are *doghouse*, *blackwood*, *redhead*, in Russian: северо-запад ‘*severo-zapad*’ (*northwest*), нефте|промышленность ‘*nefte|promyshlennost*’ (*oil industry*), цельно|металлический ‘*tsel’no|metallicheskiy*’ (*full metal*).

There are other compounds, spelled as two words or hyphenated: рыба ‘*ryba*’ (*fish*) + меч ‘*mech*’ (*sword*) → рыба-меч ‘*ryba-mech*’ (*swordfish*). This sort is getting more common because of the influx of compounds loaned (and sometimes calqued) from English. However, in this work, we classify them as pure compounds because of their structural similarity: $w_1 + w_2$.

Parasynthetic compounds are produced with both compounding and derivation with one or many affixes. The most common pattern for such words is $w_1 + w_2 + sfx$. For example, the word зелено|глазый ‘*zeleno|glazyy*’ (*green-eyed*) is a parasynthetic compound since the lexemes *зеленоглаз(а) ‘*zelenoglaz(a)*’ and *глазый ‘*glazyi*’ do not exist in the language.

According to Bisetto and Melloni (2008), words are analyzed as parasynthetic compounds in the following cases:

1. either $w_1 + w_2$ and $w_2 + sfx$ are non-existent lexemes;
2. $w_1 + w_2$ is not attested and $w_2 + sfx$ is instead an independently occurring lexeme. ‘However, scopal ambiguity effects systematically arise in this case, since the suffix has scope over the complex base, rather than just over the second stem, giving rise to a bracketing paradox’, they say.

In contrast, ‘Russkaya Grammatika’³ (Shvedova, 1980) follows a formal criterion: if the lexeme $w_2 + sfx$ exist, then a compound is pure, not parasynthetic. Thus, for example, the word древне|египетский ‘*drevne|egipetskiy*’ (*Ancient Egyptian*) is parasynthetic by the former definition and pure by the latter.

In this work, we follow the convention from ‘Russkaya Grammatika’, because it provides not only a definition but a significant number of examples for each compounding rule.

A frequent pattern in Russian and other Slavic languages is (prep + noun + sfx → adj) (cf. за ‘*za*’ (*beyond*) + граница ‘*granitsa*’ (*border*) + -н1(ый) ‘-n1(yy)’ → за|гранич|ный ‘*za|granich|nyy*’ (*overseas*) (ru), za|hranič|ní (cz), za|granicz|ny (pl)). According to the Russian linguistic tradition, the first morpheme in such words is analyzed as a prefix, hence the morphological process is derivation, not compounding.

Classical and neoclassical compounds are composed from classical Latin or ancient Greek roots: *hydrogen*, *biology*, *democracy*, *homophobia*. Such words appear in all areas of science and technology. Moreover, classical compounding is the main source of new words in these domains (Gromenko, 2020).

Some of such roots and other international morphemes (e. g. авто- ‘*avto-*’ (*auto-*), аква- ‘*akva-*’ (*aqua-*), -фобия ‘-fobiya’ (*-phobia*), мини- ‘*mini-*’ (*mini-*), etc.) can attach to normal Russian words

²<https://github.com/s231644/rucompoundsplitter>

³<http://rusgram.narod.ru/760-790.html>, § 760.

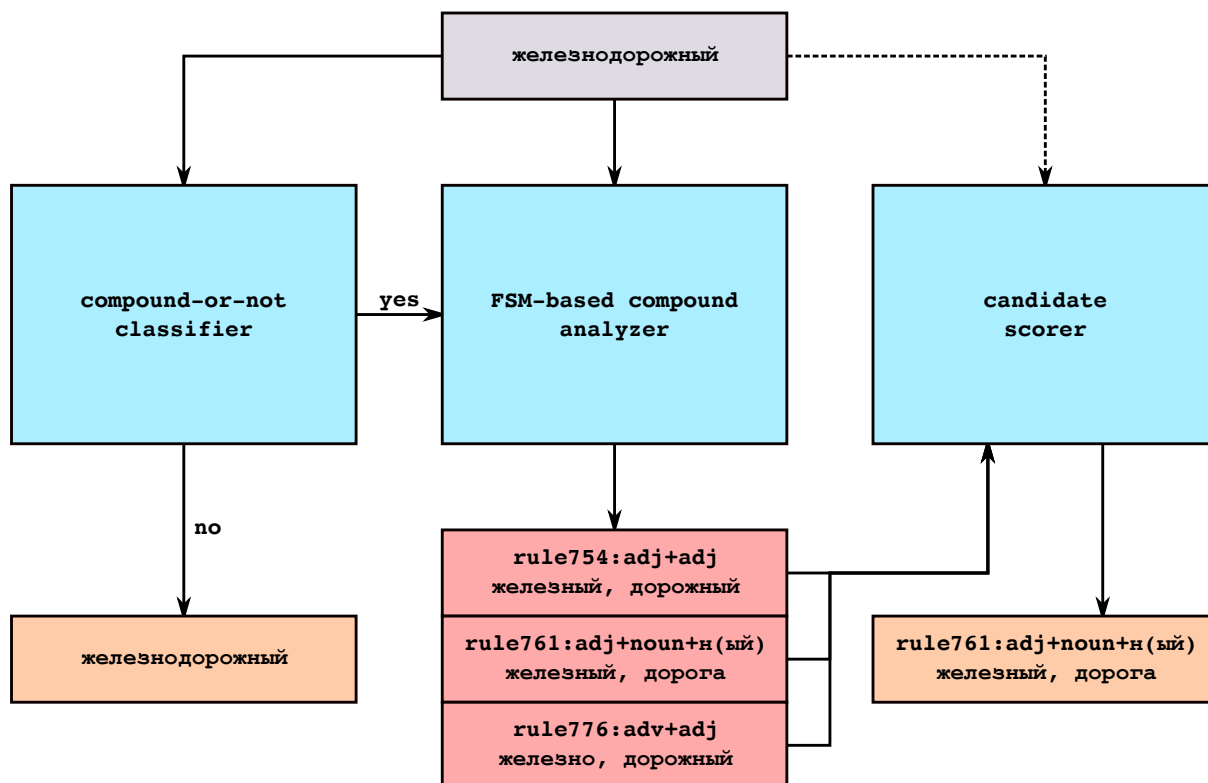


Figure 1: Diagram of the proposed pipeline. Gray: input (*zhel'eznodorozhnyy*), blue: algorithmic blocks, red: analyses generated by a rule-based model, orange: output (final analyses). The dotted line indicates that the target word is not always used in a scoring process.

acting as quasi-affixes: *игро|тека* '*igro|teka*' (*playroom*), cf. *библио|тека* '*biblio|teka*' (*library*); *мега|завод* '*mega|zavod*' (*megafactory*), but these words do not satisfy the definition of classical / neo-classical compounds.

Phrasal compounds are derived from the whole expressions. For example, the English word *above-mentioned* and its Russian counterpart *выше|упомянутый* '*vyshе|upomyanutyuy*' are phrasal compounds. Such constructions may contain more than two words like in *с|ума|сшедший* '*s|uma|sshedshiy*' (*mad, crazy, lit. gone out of mind*), or even affixes: *с|ума|сшествие* '*s|uma|sshestvie*' (*madness*) (note that **сшествие* '*sshestvie*' cannot be used as a separate lexeme), *ничего|не|делание* '*nichego|ne|delanie*' (*doing nothing*), *what|about|ism*.

The only productive phrasal compounding pattern in Russian is (adv + adj/part → adj)⁴.

3 Models

In real-world applications, we do not know if a particular word is a compound. For this reason, we separate our pipeline into three stages: compound-or-not classification, rule-based splitting and analysis, and hypotheses (candidates) scoring. Figure 1 illustrates the complete process of word analysis.

3.1 Compound-or-Not Classification

First, we classify whether a word is a compound. We solve this task as a binary classification problem with a character-level bidirectional LSTM (Hochreiter and Schmidhuber, 1997; Graves and Schmidhuber, 2005) with attention (Bahdanau et al., 2014). More formally, given a sequence of input character embeddings (including the special characters for the beginning and ending of the word) x_1, \dots, x_n , we

⁴Cf. § 778 in 'Russkaya Grammatika'.

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 segmentation⁵ from (Sorokin and Kravtsova, 2018) in this experiment. Having the morpheme segmentation for all words, we computed the number of roots in them, so the target variable in the classification task was $[\#(\text{roots}) > 1]$.

3.2 Splitting and Analysis

Next, the rule-based model is applied to produce a set of hypotheses for each word.

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)⁶. Thus, for each rule, we use the left FSM to analyze the left part and the right one to analyze the right part. Since the FSMs are the same for many rules (e. g. noun + interfix in the left part), we pre-compute all possible FSMs for the left and right parts independently and then combine the proper two for each rule.

We constructed the FSMs using the Wiktionary vocabulary. We implemented the rules for interfixes and derivational suffixes using the DerivBase.Ru framework (Vodolazsky, 2020). Many derivational suffixes that are used in parasyntetic compounding are the same as for pure derivation, so we reused the existing implementation of such rules. Then we applied all rules to all words of the corresponding part of speech and stored the produced outputs in FSMs (one FSM for one rule). Thus, for instance, for the rule **rule761**([num + ITFX] + noun + н1(ый) → adj) the left FSM has ID **ruleINTERFIX**(num) and contains all numerals with interfixes, and the right FSM with ID **rule619***(noun + н1(ый) → adj) contains all utterances obtained after adding the suffix -н1(ый) ‘-н1(yy)’ to all nouns from Wiktionary. See Figure 2 with the illustration of such FSMs. The IDs of the rules correspond to the paragraphs in ‘Russkaya Grammatika’ (Shvedova, 1980).

If a word is recognized by an FSM, then we consider such analysis as possible.

3.3 Hypotheses Scoring

Unfortunately, the described procedure is not enough, as FSMs return multiple analyses. Given a word and a set of analyses produced by a rule-based model, we score them with a model F and (optionally) select the top- k with the highest scores.

The model F scores each analysis given the left l and right r parts of the compound c .

3.3.1 Baselines

We study the following 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:

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

⁵<https://github.com/AlexeySorokin/NeuralMorphemeSegmentation>

⁶Note that the interfix and the suffix can modify the stems of the corresponding words

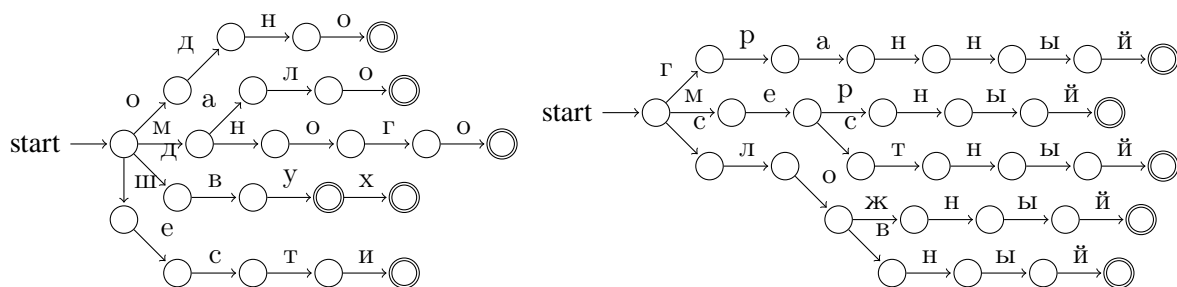


Figure 2: Left: the FSM for numerals with interfix (один ‘odin’ (one) → одно ‘odno’, два ‘dva’ (two) → дву/двух ‘dva/dvukh’, шесть ‘shest’ (six) → шести ‘shesti’, мало ‘malo’ (few) and много ‘mnogo’ (many) remain unchanged). Right: the FSM for adjectives derived from nouns грань ‘gran’ (face (of a figure)), мера ‘mera’ (measure), место ‘mesto’ (place, seat), слог ‘slog’ (syllable), слово ‘slovo’ (word) with the suffix -н(ый) ‘-n(yu)’. The concatenated FSM with ID **rule761**([num + ITFX] + noun + н1(ый) → adj) can recognize words многомерный ‘mnogomernyy’ (multidimensional), однословный ‘odnoslovnny’ (one-word), etc. The corresponding analyses would be represented in the form (**rule761**([num + ITFX] + noun + н1(ый) → adj), один, слово).

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 embeddings (Cordeiro et al., 2016):

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

We took the unigram and bigram counts from the Russian National Corpora n -gram statistics⁷. To get the counts for a paradigm, we simply sum the counts for each wordform in the paradigm. Wordforms for the given lemmas were generated with pymorphy2 (Korobov, 2015). The pre-trained ruwikiruscorpora_upos_skipgram_300_2_2019⁸ model was used for the baselines on word embeddings.

All described baselines have at least two shortcomings. Firstly, they are completely zero-shot in terms of hypotheses scoring task, but it is desirable to fit them on a ground-truth training data. On the other hand, such approach can be applied when there is no such data. Secondly, many morphemes such as Greek roots or international quasi-affixes do not typically appear as separate tokens in corpora, therefore it is difficult to correctly estimate their frequencies and impossible to retrieve their embeddings.

3.3.2 Neural Model

In contrast to the baseline models, a neural model can fit on training data. As our training dataset is relatively small, we do not expect the model to learn the necessary linguistic patterns during training. Thus, 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 (cf. Section 3.1).

The adapted to the hypotheses scoring task model takes a compound word c and N hypotheses in the form (l, r, R_l, R_r, R_c) , where l and r are as defined above, R_l and R_r are the IDs of rules applied to each of parts of the compound, R_c is the ID of the whole word-formation pattern. To process such rule IDs, we associate each of them with a trainable embedding.

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. For instance, the model will learn that word био- ‘bio-’ (bio-) is usually seen with the p_l token that corresponds to the beginning of compounds.

⁷<https://ruscorpora.ru/new/corpora-freq.html>

⁸<https://rusvectors.org/ru/models/>

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

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. Although we work here with a scoring task, we use a binary cross-entropy objective to classify whether an analysis is correct or not. During prediction, we select a candidate with the highest model’s confidence.

4 Training, Evaluation, and Error Analysis

4.1 Compound Identification

We trained the model with a binary cross-entropy objective. The model hyperparameters and the training parameters are shown in Table 1.

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 (4354 true positives, 276 false positives, 350 false negatives) on a test set.

As the numbers of false positives and false negatives were relatively small, we could interpret the model’s errors. We found that many of them can be grouped into several categories. In addition to the obvious causes of errors, such as the incorrect gold annotation, there are other reasons for them. For example, false positives could be also caused by coincidence between parts of test words and common parts of the training compounds. False negatives could be caused by language-specific phrasal abbreviations or loanwords. See Table 2 for details.

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

Table 2: Main error types in the compound-or-not classification task. Top: false positives, bottom: false negatives.

4.2 Hypotheses Generation and Scoring

We manually collected 1729 compounds with their gold-standard analyses. All compounds are taken from ‘Russkaya Grammatika’ (Shvedova, 1980). Then we split them into training (1143), validation (160), and test sets (364) preserving nearly equal distributions of rule IDs in the three sets. The gold dataset is stored as a table with the four columns: ‘rule_id’, ‘compound’, ‘left’, ‘right’. Some entries of

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

Table 3: Samples from the evaluation dataset: водопад ‘*vodopad*’ (*waterfall*), стопроцентный ‘*stoptrotsentnyy*’ (*one-hundred-percent*), травоядный ‘*travoyadnyy*’ (*herbivorous*), взаимодействовать ‘*vzaimodeystvovat*’ (*interact*).

the dataset can be found in Table 3.

For each compound, we check if a rule-based model produced the correct analysis. Given the total number of produced hypotheses, we are able to compute precision, recall and F1-measure of the rule-based model according to the following formulas:

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

The model achieved precision 0.0748, recall 0.7796, and F1-measure 0.1366 on a test set. The high recall score indicates that the proposed method works well on diverse data. The low precision score means that the model suffers from overgeneration and must be used only in combination with a scoring or filtering model. Since we do have a scoring model in this work, the most important metric for the analyzer is recall, as without producing a correct analysis, a scoring model will not be able to make a correct decision. We studied the errors of the analyzer model (false negatives) and found the main error sources listed below.

- Some of the examples from the evaluation dataset did not match the pattern $l(i)r(s)$, whereas some specific types of compounding involve prefixes (e. g. в|пол|голоса ‘*v|pol|golosa*’ (*in an undertone*)), so additional modifications needed to make it possible to analyze them with our FSMs. Another case is having more than two stems, e. g. as in газо|нефте|хранилище ‘*gazo|nefte|khranilische*’ (*oil and gas storage*). Our analysis algorithms could be modified easily, but in sake of simplicity, we did not make it in this work.
- The other source of false negatives is having the proper names and rare wordforms in one of the compound parts, e. g. чеховед ‘*chekhoved*’ (*an expert in Chekhov’s literary works*), троеборец ‘*troeborets*’ (*triathlete*) (the reference left part is трое, while the typical form of the numeral три ‘*tri*’ (*three*) is трёх ‘*tryokh*’), славянофил ‘*slavyanofil*’ (*slavophile, lit. slavs lover*), метеоусловия ‘*meteousloviya*’ (*weather conditions*) (славяне and условия are plural forms, therefore they did not appear in the lemmasets used for constructing FSMs).
- One more reason is small mistakes in the DerivBase.Ru rules, i. e. the words with zero suffixes желтощёк ‘*zheltoschyok*’ (*yellowcheek*) and шелкопряд ‘*shelkopryad*’ (*silkworm, lit. silk spinner*) were not correctly processed by the corresponding rules.
- Finally, some words in the evaluation dataset do not belong to any productive word-formation pattern and we did not implement rules for such occasional words: боеготовный ‘*boegotovnyy*’ (*combat-ready*), первогодок ‘*pervogodok*’ (*first-year*). Phrasal compounds different from (adv + adj/part → adj) also were not covered by the rules: шапкозакидательский ‘*shapkozakidatel’skiy*’ (*cocksure, lit. related to throwing caps*).

To evaluate scoring models, we use the accuracy metric, i. e. the percentage of matches of top-1 best candidates and ground-truth analyses. The best model—neural net with pre-trained weights—achieved 60.3 accuracy. The results are presented in Table 4.

Model	Accuracy
Random (100 runs)	24.89 ± 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)	60.33

Table 4: Results of the scoring models.

5 Conclusion and Future Work

In this work, we presented our approach to compound identification, splitting, and analysis for Russian. We solved this task using the combination of a rule-based method, finite state machines, and neural networks. The models we used achieved the high F1 score on a compound classification task and the high recall score on a hypotheses generation task. We evaluated and compared the five unsupervised scoring functions based on word frequencies and distributional semantics (word embeddings). The dataset used as the gold standard was manually collected and annotated by us and will be publicly available on our GitHub.

One of the directions of our future work is making an end-to-end pipeline. Also, we aim to try different model architectures such as convolutional neural networks or transformers instead of BiLSTMs. We hope that this will improve the quality of compound analysis.

The second direction is an application of the proposed algorithm to a large-scale vocabulary and integration of compounds into the Russian part of Universal Derivations (Kyjánek et al., 2019).

Finally, we plan to collect a larger dataset for training and evaluation of compound analysis models.

Acknowledgments

We thank Igor Galitskiy for providing his computing server and the anonymous reviewers for their comments and suggestions.

References

- Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.
- Antonietta Bisetto and Chiara Melloni. 2008. Parasyntetic compounding. *Lingue e linguaggio* 7(2):233–260.
- Silvio Cordeiro, Carlos Ramisch, Marco Idiart, and Aline Villavicencio. 2016. Predicting the compositionality of nominal compounds: Giving word embeddings a hard time. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Berlin, Germany, pages 1986–1997. <https://doi.org/10.18653/v1/P16-1187>.
- Alex Graves and Jürgen Schmidhuber. 2005. Framewise phoneme classification with bidirectional lstm and other neural network architectures. *Neural networks* 18(5-6):602–610.
- ES Gromenko. 2020. Special'naja leksika xx veka kak ob'ekt neografii: K istorii voprosa (na materiale slovarja-spravočnika po materialam pressy i literatury 60-h gg.). *Vestnik Nižegorodskogo universiteta im. NI Lobačevskogo* (3).
- Verena Henrich and Erhard Hinrichs. 2011. Determining immediate constituents of compounds in GermaNet. In *Proceedings of the International Conference Recent Advances in Natural Language Processing 2011*. Association for Computational Linguistics, Hissar, Bulgaria, pages 420–426. <https://www.aclweb.org/anthology/R11-1058>.

- Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9(8):1735–1780.
- Valentin Hofmann, Janet B. Pierrehumbert, and Hinrich Schütze. 2021. Superbizarre is not superb: Derivational morphology improves bert’s interpretation of complex words.
- Philipp Koehn and Kevin Knight. 2003. [Empirical methods for compound splitting](#). In *10th Conference of the European Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, Budapest, Hungary. <https://www.aclweb.org/anthology/E03-1076>.
- Mikhail Korobov. 2015. Morphological analyzer and generator for russian and ukrainian languages. In *Analysis of Images, Social Networks and Texts*, Springer International Publishing, volume 542 of *Communications in Computer and Information Science*, pages 320–332.
- Lukáš Kyjánek, Zdeněk Žabokrtský, Magda Ševčíková, and Jonáš Vidra. 2019. Universal derivations kickoff: A collection of harmonized derivational resources for eleven languages. In *Proceedings of the Second International Workshop on Resources and Tools for Derivational Morphology*. pages 101–110.
- Haonan Li, Zhisong Zhang, Yuqi Ju, and Hai Zhao. 2018. Neural character-level dependency parsing for Chinese. In *Thirty-Second AAAI Conference on Artificial Intelligence*.
- Helmut Schmid, Arne Fitschen, and Ulrich Heid. 2004. [SMOR: A German computational morphology covering derivation, composition and inflection](#). In *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC’04)*. European Language Resources Association (ELRA), Lisbon, Portugal. <http://www.lrec-conf.org/proceedings/lrec2004/pdf/468.pdf>.
- Rico Sennrich and Barry Haddow. 2015. A joint dependency model of morphological and syntactic structure for statistical machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. pages 2081–2087.
- Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*.
- Natalija Shvedova. 1980. *Russkaja grammatika*. Number t. 1 in *Russkaja grammatika*. Izd-vo Nauka. <https://books.google.ru/books?id=7BNgAAAAMAAJ>.
- Alexey Sorokin and Anastasia Kravtsova. 2018. Deep convolutional networks for supervised morpheme segmentation of russian language. In Dmitry Ustalov, Andrey Filchenkov, Lidia Pivovarova, and Jan Žižka, editors, *Artificial Intelligence and Natural Language*. Springer International Publishing, Cham, pages 3–10.
- Bogdan Szymanek. 2017. Compounding in Polish and the absence of phrasal compounding. *Further investigations into the nature of phrasal compounding* 1:49.
- Daniil Vodolazsky. 2020. [DerivBase.Ru: a derivational morphology resource for Russian](#). In *Proceedings of the 12th Language Resources and Evaluation Conference*. European Language Resources Association, Marseille, France, pages 3937–3943. <https://www.aclweb.org/anthology/2020.lrec-1.485>.
- Marion Weller-Di Marco. 2017. [Simple compound splitting for German](#). In *Proceedings of the 13th Workshop on Multiword Expressions (MWE 2017)*. Association for Computational Linguistics, Valencia, Spain, pages 161–166. <https://doi.org/10.18653/v1/W17-1722>.