# АВТОМАТИЧЕСКАЯ СЕГМЕНТАЦИЯ СЛОЖНЫХ СЛОВ ПУТЕМ КОМБИНИРОВАНИЯ ЯЗЫКОЗАВИСИМЫХ И ЯЗЫКОНЕЗАВИСИМЫХ ПРИЗНАКОВ

**Логинова-Клуэ Е. А.** (elizaveta.loginova@univ-nantes.fr),
**Daille B.** (beatrice.daille@univ-nantes.fr)

Университет г. Нант, Нант, Франция

**Ключевые слова:** сегментация сложных слов, меры близости, правила трансформации компонентов, специализированный корпус

# MULTILINGUAL COMPOUND SPLITTING COMBINING LANGUAGE DEPENDENT AND INDEPENDENT FEATURES

**Loginova-Clouet E. A.** (elizaveta.loginova@univ-nantes.fr),
**Daille B.** (beatrice.daille@univ-nantes.fr)

Nantes University, Nantes, France

Compounding is a common phenomenon for many languages, especially those with rich morphology. Dealing with compounds is a challenge for NLP systems since compounds are not often included in the dictionaries and other lexical sources. We present a compound splitting method combining language independent features (similarity measure, corpus data) and language specific component transformation rules. Due to the usage of language independent features, the method can be applied to different languages. We report on our experiments in splitting of German and Russian compound words, giving positive results compared to matching of compound parts in a lexicon. To the best of our knowledge. elaborated compound splitting is a rare component of NLP systems for Russian, yet our experiments show that it could be beneficial to use a specialized vocabulary.

**Key-words:** compound splitting, multilingual tool, similarity measure, component transformation rules, specialized corpora

## 1. Introduction

Compounding is a method of word formation consisting in a combination of two (or more) autonomous lexical elements that form a unit of meaning. This phenomenon is common in German, Dutch, Greek, Swedish, Danish, Finnish and many other languages. In Russian compounding is less regular, but also present, especially in specialized fields. Compound treatment is a problem for the automatic NLP systems because most of compounds are not listed in lexical sources, and not so frequent to be observed in training data. However, their recognition and splitting could be of benefit for various NLP tasks (machine translation, information retrieval, terminology extraction, etc.).

Compounding mechanisms are more or less complex depending on language. In highly analytical languages such as English or French, compound parts are just concatenated: EN[1] *parrotfish*, FR *kilowatt-heure*, kilowatt-hour. In languages with a rich morphology, some transformations are possible at the boundary of the compound parts. The word ending can be omitted, and/or boundary morphemes can be added, for example in DE:

(1) *Staat**s**feind* (state enemy) = *Staat* (state) + *Feind* (enemy)*;*

(2) *Muse**en**verwaltung (museum administration) = Museum (museum) + Verwaltung (administration);*

For some languages the list of such rules is rather short and exhaustive, for others it is more complicated. Sometimes a modification of the stem is possible, as in Russian:

(3) *ве**тро**генератор (wind generator) = ве**тер** (wind) + генератор (generator);*

A special case appears with the "neoclassical compounds", i.e. compounds which one element or more has Latin or Greek etymological origin [Namer, 2009]. For example EN *multimedia*, DE *turbomaschine* (turbomachine), etc. Usually these elements are not autonomous, but represent the units of meaning. Sometimes neoclassical elements are included in dictionaries or lexical databases.

In this paper, we examine some existing methods of compound splitting and then we propose our method combining language dependent and independent features. We report on our experiments in splitting German and Russian compounds. We conclude with some remarks on compound splitting in general and on its particularity in Russian language.

[1] EN — English language, FR — French language, DE — German language, RU — Russian language

## 2. Compound Splitting Methods

Compound splitting methods can be divided into supervised (generally rule-based) and unsupervised (fully statistical) methods. Let us illustrate the first type of methods on the example of German compound splitters. These are often based on the study of [Langer, 1998], describing the transformation rules for compound formation in this language. Systems check whether a word's component matching with a dictionary [Ott, 2005] or with a monolingual corpus [Koehn and Knight, 2003], [Weller and Heid, 2012]. Corpus-based approaches give also a probability for each segmentation, estimated from the components frequencies in the corpus. A parallel English corpus could be involved to check correspondences of decomposed parts [Koehn and Knight, 2003]. These methods are robust and give high results for the languages they were designed for.

The second group of approaches are language independent. [Macherey et al., 2011] propose to automatically extract morphological operations at the components boundary. The training of the model for a new language requires a parallel English corpus. It allows the authors to test their method for several languages: Danish, German, Norwegian, Swedish, Greek, Estonian, Finnish. [Hewlett and Cohen, 2011] detect automatically the places of components boundaries. The algorithm is based on the probability of the character's sequences in a language. The measure of probability is entropy: the entropy inside the word is relatively low, whereas the entropy on the word/component boundaries is much higher. Currently fully statistical models are not as precise as rule-based methods, but their advantage is their reusability for any language.

## 3. Multilingual Compound Splitting Algorithm

Our goal was to design a compound splitting tool that could be applied to different languages through language independent features, but also able to integrate linguistic rules if they are available for a given language. As a language independent feature, we chose monolingual corpus data and similarity measure between a word subsequence and the candidate lemmas.

To split a compound, we start with forming all its possible two-part segmentations beginning with the components of minimum permitted length (which is a parameter):

(4) *DE Magnetisierungszustand (magnetisation state)*
*magnetisierungszustand -> ma + gnetisierungszustand*
*magnetisierungszustand -> mag + netisierungszustand*
…
*magnetisierungszustand -> magnetisierungszusta + nd*

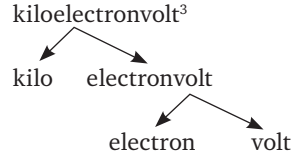
If the specific rules for component transformation into independent lexemes are available for this language, we apply them to the candidate components before matching with the dictionary/corpus. These are the rules of type: "s" → " " cf. (1), "en" → "um" cf. (2), etc.

For each candidate segmentation, both parts are matched with a monolingual dictionary, and optionally with a monolingual corpus. The corpus serves to calculate words frequency, which enables the tool to choose more plausible component candidates if several variants are possible. The corpus should be of reasonable size to obtain realistic distribution of words. The corpus is particularly useful if we deal with specialized vocabulary, containing many highly specialized terms not described in general language dictionaries.

If we do not have transformation rules or if they do not let finding a lemma, we exploit similarity measure. When searching in the dictionary/corpus, we calculate similarity between the segmentation part and candidate lemmas to choose the "closest" lemmas[2]. Various similarity measures could be used. So far we tried "normalized edit distance", based on Levenshtein distance, and "longest common prefix" measures (for detailed outlook of existing measures see [Frunza and Inkpen, 2009]).

If some acceptable lemmas are found for the left part of the current segmentation, but not for the right part, we try to split further the right part in a recursive manner, and so on up to a certain level. This level is a parameter corresponding to the maximum number of components.

RU килоэлектронвольт: kiloelectronvolt[3]




If we have acceptable candidate lemmas for all components, we calculate the score for this segmentation based on obtained similarity value, existence in the dictionary and (optionally) frequency in the corpus for each component. Finally, the tool returns a top N of the best segmentations ordered by their score. For example, for DE Magnetisierungszustand (magnetisation state) the output is:

| | |
|---|---|
| magnetisierung + zustand | 2.00 |
| magnete + sie + erregungszustand | 0.75 |
| magnete + sicherungskasten[4] | 0.69 |

The correct split is Magnetisierung (magnetisation) + Zustand (state), and it has the best score given by the program. The algorithm enables to set various parameters depending on our heuristics for a given language and on the application aimed. The user chooses either to split all given words (it supposes that all given words are compounds), either to first match each word with a dictionary and not to split the words found (the case of application to machine translation). Source code with detailed algorithm description, as well as test data, are available online[5].

---

[2] The threshold for lemma acceptance is a parameter

[3] Russian word is given in transliterated form

[4] The word "sicherungskasten" has similarity of 0,6 with the substring "sierungszustand", that is why it was found by the program

[5] http://www.lina.univ-nantes.fr/?Compound-Splitting-Tool.html

## 4. Experiments

In this section we report on our experiments using the algorithm described above. So far we applied it for compound splitting in two languages, German and Russian. We chose German because in this language compounding is very productive and well-described. Compounding in Russian is less frequent, so the question can be asked: does an NLP system for Russian really need a splitting mechanism, or is it sufficient just to add all known components in the system lexicon? Our experiments were guided by this question.

For both languages, we analyzed compound words from the domain of wind energy. We varied some parameters to observe the impact of corpus usage, of boundary transformation rules and similarity measure on the quality of splitting. As a baseline we performed splitting only with a dictionary, as if we were simply searching for the word components in the lexicon (that is applied in some NLP systems).

To evaluate the results, we calculated precision at rank 1 (top 1) and precision at rank 5 (top 5). Precision is calculated as the number of correct splits divided by the total number of compounds. We did not calculate recall in these experiments because we only analyzed compound words. A procedure deciding to split a token or not can constitute a topic for future researches.

### 4.1. Experiments with German compounds

For German language, three experiments were done: baseline splitting (only with a dictionary); splitting with dictionary and boundary transformation rules; and splitting with dictionary, rules and corpus filtering. The rules used in the second and third experiments are based on [Langer, 1998] work. Similarity is based on Levenshtein distance measure.

We used a German part of free German-English dictionary Dict. cc[6] (800,000 word entries); a specialized corpus related to wind energy domain crawled from the web[7] (300,000 words); and a test set of 446 compounds for splitting[8] consisting of two, three or four components. The results are presented in Tab. 1.

**Table 1.** Splitting Precision for German Language

| | Baseline | Rules, no corpus | Rules, corpus |
|---|---|---|---|
| Top 1 | 66.59% | **93.04%** | 87.44% |
| Top 5 | 66.59% | 95.06% | **95.51%** |

[6] http://www.dict.cc

[7] http://www.lina.univ-nantes.fr/?Linguistic-Resources-from-the.html

[8] [Weller and Heid, 2012], data available at: http://www.ims.uni-stuttgart.de/~wellermn/tools.html

The results with addition of transformation rules and similarity measure are clearly better than those of baseline-experiment. The results are comparable with those given by methods designed for German language: thus, [Koehn and Knight, 2003] report on accuracy of 95,7%[9] for their monolingual frequency based method.

The usage of corpus slightly increases precision for top 5. It allows a correct splitting of some additional words, whose components are not present in the dictionary (Netzanschluß, "network connection"). In some cases, it also improves the ranking: Traktionsbatterie without corpus returns two equal-ranked splits traktion + batterie 1.0 and trakt + ion + batterie 1.0. The usage of corpus raises the correct split: traktion + batterie 1.50, trakt + ion + batterie 1.25. Though, in other cases corpus affects the ranking because it promotes the splits consisting in shorter and more frequent components: Aberrationswinkel, "aberration angle", without corpus is correctly split in aberration + winkel, and with corpus the best-ranked split is aber + ration + winkel. That is why the precision in top 1 with corpus is lower than without. This problem may be resolved in replacing simple corpus frequency by specificity, i.e. comparing special corpus frequency to the frequency in a general corpus (cf. "weirdness ratio" [Ahmad et al., 1992]).

### 4.2. Experiments with Russian compounds

For Russian language, in addition to baseline experiment, we varied three parameters: the usage or not of the corpus; similarity measure (Levenshtein distance vs. The longest common prefix, later "Prefix"); and small rules-set or large rules-set. So we did 9 experiments with different combinations of these parameters.

The transformation rules for Russian were formulated on the base of description of Russian morphology in [Zaliznjak, 1977]. The small rules-set consists in two simple rules expressing the common knowledge that linking morphemes "о" and "е" operate as boundary morphemes in Russian. For the full rules-set see Table 2.

**Table 2.** Transformation Rules for Russian compounds

| N | Left context | Transformation | Example |
|---|---|---|---|
| **Small rules-set** | | | |
| 1 | - | "о" → " " | |
| 2 | - | "е" → " " | |
| **Large rules-set** | | | |
| 3 | - | "о" → "а" | водо- / вода |
| 4 | - | "е" → "я" | земле- / земля |
| 5 | "ж"\|"ш"\|"щ"\|"ч"\|"ц" | "е" → "а" | тысяче- / тысяча |
| 6 | - | "е" → "ь" | жизне- / жизнь |
| 7 | - | "о" → "ый" | крупно- / крупный |

[9] Accuracy is calculated here in the same way we calculate precision.

| N | Left context | Transformation | Example |
|---|---|---|---|
| 8 | - | “о” → “ой” | криво- / кривой |
| 9 | - | “е” → “ий” | обще- / общий |
| 10 | “к”\|“г” | “о” → “ий” | высоко- / высокий |
| **Inflexion rules** | | | |
| 11 | - | “ый” → “ ” | |
| 12 | - | “ий” → “ ” | |
| 13 | - | “ой” → “ ” | |

We used electronic version of [Ozhegov, 1991] dictionary (nearly 61,000 words), completed by a list of neoclassical elements taken from [Béchade, 1992] and translated into Russian. Neoclassical elements are very frequent in Russian compounds and necessary for correct splitting, but only few of them were already included in the dictionary. We used a Russian monolingual wind energy corpus of 300,000 words crawled from the web[10].

Test data are issued from the wind energy corpus. Among 7,000 most frequent lexemes in this corpus, 348 are compounds. It confirms that compounding in Russian, even if it is not as productive as in some Germanic languages, needs to be taken into account, at least for specialized domains. The results for all compounds are presented in Table 3.

**Table 3.** Splitting Precision for Russian Language

| | Base-line | Levenshtein, no corpus | | Levenshtein, corpus | | Prefix, no corpus | | Prefix, corpus | |
|---|---|---|---|---|---|---|---|---|---|
| | | Small rules | Large rules | Small rules | Large rules | Small rules | Large rules | Small rules | Large rules |
| Top 1 | 35.06% | 62.64% | 75.57% | 76.44% | 84.77% | 58.05% | 68.97% | 72.99% | 78.74% |
| Top 5 | 35.06% | 71.84% | 81.32% | 86.78% | 92.82% | 69.83% | 80.17% | 90.52% | 92.24% |

We noted a significant difference between baseline and other results. The usage of corpus was definitely beneficial in all analyzed cases. Some component lemmas were not present in the dictionary (*дизель*, diesel, *интернет*, internet, etc.). Compounds containing these components were correctly split through the corpus. In small rules set experiments, the Prefix similarity measure was a bit better for top 5. In some cases this measure compensates for the absence of inflexion treatment because it compares the common beginning of strings. However, the addition of large rules allowed inflexion treatment, and the measure based on Levenshtein distance became more efficient for top 1 and 5.

The impact of large rules set was not spectacular for top 5 with the usage of corpus, since for certain compounds the corpus compensates for the lack of rules. For example, the adjective *электромагнитный* (electromagnetic) could not be correctly split just with a baseline-method because its right component *магнитный* (magnetic)

[10] http://www.lina.univ-nantes.fr/?Linguistic-Resources-from-the.html

is not in the dictionary. It could be correctly split either with the usage of corpus (where *magnetic* is present and has a relatively high frequency), either with the transformation rules which enable to find the noun *магнит,* magnet:

(5) *электромагнитный → электро + магнитный*
*rule 11 (магнитный) = магнитн*
*similarity (магнитн, магнит) = 0,86*
*result : электромагнитный = электро + магнит*

By contrast, we noted a good improvement through the large rules for top 1 with the corpus (6–8 % increasing of precision), and also for all experiments without corpus (10–13 % increasing of precision).

## 5. Conclusion

We have presented a compound splitting algorithm combining language independent features (similarity measures, word frequencies in a corpus) with language dependent features (component boundary transformation rules). For the two analyzed languages, this mechanism outperforms a baseline method, consisting in a matching of the word components in a dictionary. The usage of a specialized corpus allows us to correctly split some additional compounds including components unknown in a dictionary, and enables to a certain extent to compensate the lack of transformation rules. Using more rules enables although to achieve better ranking of splits. The algorithm can be applied to other languages by changing the lexical sources and, optionally, editing transformation rules.

Concerning compound splitting in Russian, it seems to deserve a special treatment in the NLP systems, at least for the systems dealing with specialized texts. Another solution, currently used in some systems, is to keep all compound components in the lexicon, which largely increases lexicon size. This solution does not seem satisfactory for multilingual systems since it requires to complete the lexicon for each new language.

## 6. Acknowledgements

The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007–2013) under grant agreement no. 248005.

## References


1. *Ahmad K., Davies A., Fulford H. and Rogers M.* (1992), "What is a term? The semi-automatic extraction of terms from text". Translation Studies: An Interdiscipline, John Benjamins, Amsterdam/Philadelphia, pp. 267–278.
2. *Béchade H.-D.* (1992), Phonétique et morphologie du français moderne et contemporain, Presses Universitaires de France, Paris.
3. *Frunza O., Inkpen D.* (2009), "Identification and Disambiguation of Cognates, False Friends, and Partial Cognates Using Machine Learning Techniques". International Journal of Linguistics, Vol. 1, No. 1, available at: http://www.macrothink.org/journal/index.php/ijl/article/view/309/193
4. *Hewlett D., Cohen P.* Fully Unsupervised Word Segmentation with BVE and MDL. Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics (ACL), pp. 540–545, Portland, Oregon, 2011.
5. *Koehn, P., Knight, K.* Empirical methods for compound splitting. Proceedings of EAC-2003, Budapest, Hungary.
6. *Langer, S.* Zur Morphologie und Semantik von Nominalkomposita. Proceedings of 4th Conference Computers, linguistics and phonetics between language and speech (KONVENS) , Bonn, 1998, pp. 83–97.
7. *Macherey K., Dai A. M., Talbot D., Popat A. C., Och F.* Language-independent Compound Splitting with Morphological Operations. Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics (ACL). Portland, Oregon, 2011, pp. 1395–1404.
8. *Namer Fiammetta* (2009), Morphologie, lexique et traitement automatique des langues, Lavoisier, Paris.
9. *Ozhegov S. I.* (1991), Tolkovyj slovar' russkogo jazyka [Russian Language Dictionary], web version available at: http://speakrus.ru/dict/ozhegovw.zip.
10. *Ott, N.* (2005), "Measuring Semantic Relatedness of German Compounds using GermaNet", available at: http://niels.drni.de/n3files/bananasplit/Compound-GermaNet-Slides.pdf
11. *Weller M., Heid U.* Analyzing and Aligning German Compound Nouns. Proceedings of the 8th International Conference on Language Resources and Evaluation (LREC), Istanbul, 2012.
12. *Zaliznjak, A. A.* (1977), Grammaticheskij Slovar' Russkogo Jazyka [Grammatical Dictionary of the Russian Language], Russkij jazyk, Moscow.