Semi-automatic construction of word-formation networks

Mateusz Lango¹ · Zdeněk Žabokrtský² ·
Magda Ševčíková²

Published online: 23 January 2020
© The Author(s) 2020

Abstract The article presents a semi-automatic method for the construction of word-formation networks focusing particularly on derivation. The proposed approach applies a sequential pattern mining technique to construct useful morphological features in an unsupervised manner. The features take the form of regular expressions and later they are used to feed a machine-learned ranking model. The network is constructed by applying the learned model to sort the lists of possible base words and selecting the most probable ones. This approach, besides relatively small training set and a lexicon, does not require any additional language resources such as a list of vowel and consonant alternations, part-of-speech tags etc. The proposed approach is evaluated on lexeme sets of four languages, namely Polish, Spanish, Czech, and French. The conducted experiments demonstrate the ability of the proposed method to construct linguistically adequate word-formation networks from small training sets. Furthermore, the performed feasibility study shows that the method can further benefit from the interaction with a human language expert within the active learning framework.

Keywords Derivation · Derivational morphology · Word-formation · Lexical network · Learning to rank · Sequential pattern mining · Active learning

✉ Mateusz Lango
mateusz.lango@cs.put.poznan.pl
Zdeněk Žabokrtský
zabokrtsky@ufal.mff.cuni.cz
Magda Ševčíková
sevcikova@ufal.mff.cuni.cz

¹ Faculty of Computing, Institute of Computing Science, Poznan University of Technology, Poznan, Poland

² Faculty of Mathematics and Physics, Institute of Formal and Applied Linguistics, Charles University, Prague, Czech Republic

1 Introduction

1.1 Motivation

Derivational morphology has moved into the focus of Natural Language Processing (NLP) rather recently. In the last decade, we observe a significant research effort in the construction of resources specialized in derivation for a number of languages. However, the creation of such resources requires considerable human effort and is highly time-consuming. As there are still many languages for which data resources providing information about derivation are scarce or even lacking, more efficient and less costly approaches should be sought, like in other areas of NLP and Computational Linguistics.

1.2 Linguistic decisions taken

The presented approach of creating word-formation networks is currently limited to derivation, which is a word-formation process consisting in adding an affix morpheme to a root morpheme or to an existing word (base lexeme) so that a new word (derived lexeme, derivative) is created this way. Derivation is attested across languages all over the world, being the major process in Slavic languages (incl. Czech and Polish analyzed here) and playing an important role also in other European languages (incl. Spanish and French); cf. Štekauer et al. (2012) typological survey of word-formation in a sample of 55 languages from 28 language families, or Körtvélyessy (2016) on word-formation in Slavic languages and Körtvélyessy et al. (2018) on European languages.

While one can see manifestations of derivation in natural languages relatively easily on the common sense level, it is less clear how this phenomenon should be optimally modelled and what abstract data structure model should be used. Several different formalizations have been used in the relevant literature (see Sect. 2.1). The one we adopt in this article models derivation as a binary relation between the base lexeme and the derivative; it can be described by using elementary notions of graph theory as follows:

- lexemes present in the vocabulary of a particular language under study are represented as nodes in an oriented graph,
- if we recognize a pair of lexemes that share the same root morpheme to be immediately derivationally related, then an oriented edge is inserted into the graph; the edge represents the derivation from the base lexeme to the derived lexeme,¹
- all derivationally related lexemes (sharing a common root morpheme) constitute a derivational cluster, also called a derivational family, having the form of a rooted tree (hereafter, derivational tree),
- the whole graph (formally a forest) composed of individual rooted trees is referred to as derivational network in our article. A more general term “word-

¹ Alternatively, the terms parent lexeme or antecedent lexeme, and child lexeme, descendant lexeme or derivative, respectively, are used, too.

Fig. 1 The derivational tree with the base noun *kachna* ‘duck’ as the root node (from the DeriNet network for Czech). The node labels include the part-of-speech category (marked with capital letters: N for nouns, A for adjectives, V for verbs, and D for adverbs) and possibly also the number of further descendants not displayed here (in parentheses)

formation network” is used in the same meaning (pointing to a potential broadening of the scope of the representation to other word-formation processes which is, nevertheless, not addressed in the article).

Concurring with the linguistic account of derivation as a basically affix-adding procedure (see, for instance, Dokulil 1962; Iacobini 2000; Lieber and Štekauer 2014), the lexemes within each derivational tree are organized according to their morphemic and semantic complexity from the formally simplest and semantically broadest unmotivated lexeme, which is represented by the root node, to those with the richest morphemic structures and most specific meanings in the leaves of the tree. Cf. the sample tree from the DeriNet derivational network for Czech (Ševčíková and Žabokrtský 2014) in Fig. 1, in which the noun *kachna* ‘duck’ is the base word for (among others) the young animal noun *kachně* ‘young duck’ which, in turn, becomes the base for the diminutive *kachňátko* ‘duckling’.

As confirmed in the DeriNet network, this representation suits the vast majority of derivations identified in the dictionary of contemporary Czech. Still, there are base-derivative pairs that violate this assumption, such as zero-derived action nouns (e.g., *běh* ‘run’) based on verbs that are longer (*běhat* ‘to run’) in Czech. Another example that does not comply with the basic definition of derivation are lexemes derived by conversion without adding any linguistic material. For instance, the Czech noun *cestující* ‘passenger’ is based on the formally identical adjective, or English noun *run* is converted from the formally same verb, the direction of derivation in both pairs being suggested in compliance with lexicographic resources. Such cases, which are very difficult to process automatically because of requiring a non-trivial amount of manual annotation with deep linguistic insights, have to be neglected within the approach introduced in the present article, which may affect the performance of the proposed approach when evaluated on hand-annotated data.

Another issue which one can face when dealing with real language data is that the derivational family does not contain an unmotivated lexeme because, for instance, it is not available in synchrony, though being attested diachronically. In such families, the morphematically simplest lexeme is to be chosen for the root of the tree structure.

Last but not least, the proposed representation seems to be suitable to model derivation, but it cannot be applied to compounding since more than one base lexeme needs to be identified. For this purpose, the representation has to be modified substantially, which goes, however, beyond the scope of the present article.

Being aware of these limitations, rooted trees are used here as a simple yet highly constrained data structure which makes it possible to organize massive amounts of language data in a unified way.

1.3 Task statement and overview of our approach

Our goal is to find a method for semi-automatic construction of derivational networks that can be applied to under-resourced languages, with as few input resources as possible. In this article, we propose an approach based on machine learning which requires only two resources: a set of lexemes and a relatively small training set containing examples of derived lexemes with their base words.

First, the method is looking for frequent patterns in the lexeme set in order to automatically detect grapheme-level regularities in the lexemes of the language under consideration. The mentioned process is conducted in a completely unsupervised manner, and no hand-crafted rules are used. Given those frequent patterns, each lexeme can be described by the presence (or absence) of a given pattern. Such lexeme descriptions serve as feature vectors for (supervised) machine learning techniques and allow to train a ranking model aimed at selecting the proper base lexeme for a given derived lexeme. The ranker is trained using small hand-annotated data. In the prediction phase, for each lexeme the candidate base lexeme with the best predicted ranking score is chosen, as long as it is clearly superior with respect to the second best candidate. Due to the fact that derivational edges are inserted independently of each other, the resulting graph must be post-processed in order to remove possibly generated oriented cycles.

We evaluate our approach on Polish, Spanish, Czech, and French. We are not aware of any publicly available derivational resource for Polish and Spanish, so we consider these two languages truly under-resourced from the viewpoint of our task. On the contrary, Czech and French possess well elaborated derivational resources, but we choose these languages as they allow to verify our approach on large evaluation data. In addition, the selection of the languages was motivated by the fact that Polish and Czech are both Slavic languages, while Spanish and French are Romance languages; thus we hope to be able to find some cross-lingual patterns in the future.²

1.4 Structure of the article

The rest of the article is organized as follows. In Sect. 2 we briefly present related work. Section 3 contains a description of the proposed approach based on combining two machine learning techniques. Section 4 presents gathered datasets

² However, we admit that the choice of languages under this study was naturally influenced also by the language competence of the authors of this article.

used for training and evaluation purposes in our experiments. In Sect. 5 we proceed with the discussion of experimental evaluation and with the analysis of the resulting resources. Section 6 discusses advantages and disadvantages of our machine learning approach. Finally, in Sect. 7 we draw conclusions and discuss lines of future research.

2 Related work

2.1 Existing language resources for word-formation

Most language resources that focus on word-formation, or even more specifically on derivation, have been created in the last decade, still with a handful of pioneering exceptions such as CELEX (Baayen et al. 1996). The resources cover several European languages, cf. DerIvaTario for Italian (Talamo et al. 2017), Word Formation Latin (Litta et al. 2016), CatVar (Habash and Dorr 2003) for English, DeriNet (Ševčíková and Žabokrtský 2014; Žabokrtský et al. 2016) and Derivancze (Pala and Šmerk 2015) for Czech, DerivBase.hr (Šnajder 2014) and CroDeriV (Šojat et al. 2014) for Croatian, DErivCelex (Shafaei et al. 2017) and DErivBase for German (Zeller et al. 2013), and resources created by a language-independent approach by Baranes and Sagot (2014). For some languages, even more different resources are available. For instance, Framorpho-FR (Hathout 2005) and Démonette (Hathout and Namer 2014) exist for French; Démonette is closely related to Morphonette (Hathout 2010) and VerbAction (Hathout et al. 2002).

The resources differ in many aspects, from those concerning the theoretical background and basic design decisions to rather technical ones, for instance:

- the scope of word-formation processes captured in the resource, i.e., whether only derivation as a prototypically binary relation (between a derivative and a single base word) is modelled, or other word-formation processes are involved too; the other processes include especially compounding and processes that combine derivation and compounding,
- the data structure chosen for the representation of the lexemes and relations among them; in addition to rooted trees, which are taken as a default representation of derivational families in the present article (cf. Sects. 1.2 and 4), the following options can be observed in the existing resources:
 - in some resources such as DerivBase.hr, lexemes belonging to a particular derivational family are simply grouped together, leaving concrete derivational relations within the groups underspecified; such a partitioning could be represented as a graph composed of components (connected subgraphs, each of them representing one family), however, given that there are no distinguished lexemes in particular families, the only completely symmetric graph representation of a family is a complete graph, which is rather inefficient,
 - a weakly connected subgraph (in which there exists an undirected path between any two nodes) can be used for representing derivational families in resources in which the rooted-tree constraint does not hold, e.g., in Démonette,

- a derivation tree (in the terminology of Context-Free Grammars), with morphemes in its leaf nodes and artificial symbols in non-terminal nodes, can be used for describing how a lexeme is composed of individual morphemes (cf. CELEX); derivational relations between lexemes are then present only implicitly (based on shared sequences of morphemes),
- paradigms theoretically backgrounded in the discussion summarized by Bonami and Strnadová (2018)
- distributional properties of the lexeme set in terms of part-of-speech categories, attestedness in corpus data, inclusion of potential formations, archaic lexemes etc. (e.g., large-coverage resources like CELEX vs. CroDeriV focusing on verbs),
- the general purpose, i.e., whether a resource was designed primarily for linguistic research (cf. CELEX in psycholinguistic research) or rather for NLP tasks (e.g., CatVar),
- the method of creation (usage of manual or automatic procedures, or a combination of both),
- the size of the data, reaching from 11 thousand lexemes in DerIvaTario to one million lexemes in DeriNet,
- technical accessibility, since many of the resources are available for download while others can be just queried using a web form (Derivancze, Unimorph), and last but not least
- legal availability for users, since different licenses apply to individual resources.

In addition to the resources specialized in derivational morphology, some kind of derivational information can also be found in resources of other types, namely in some corpora and lexical resources not focusing primarily on derivations. A preliminary approach to a small set of most regular derivations was implemented in the Prague Dependency Treebank 2.0 (Hajič et al. 2006) as a part of the deep-syntactic annotation. In the Russian National Corpus, selected derivatives are assigned labels capturing basic semantic categories added through derivation (such as diminutive, augmentative), however, without reference to the base lexeme (Apresjan et al. 2006).

Sets of derivational labels are also available in WordNet databases for selected languages, e.g., for English (Fellbaum et al. 2007) and Czech (Pala and Hlaváčková 2007); nevertheless, derivational annotation is not consistent across different wordnets. The last type of lexical resources to be mentioned in relation to derivations here are lexicons of verb-noun relations focusing on nominalizations, such as Nomage (Balvet et al. 2010) for French, and Nomlex-PT (de Paiva et al. 2014) for Portuguese. A comprehensive comparison of existing resources for more than 20 languages with a detailed summary of formats and licenses can be found in Kyjánek (2018).

Even when taking into account derivational information in the above mentioned types of data, derivational morphology is still strongly under-resourced when compared to inflectional morphology (cf. the recent shared tasks on inflections; Cotterell et al. 2017, 2018). For most languages, though, such comparison is not feasible at all. For instance, there is, to the best of our knowledge, no word-formation network for two languages which are considered in the present work,

Polish and Spanish. For those languages, the number of works on automatic detection of derivatives is also quite limited.

2.2 Employed machine learning techniques

Even a most humble overview of machine learning methods used for building morphological resources would go beyond the scope of this article. We limit ourselves to a description of the three techniques that we use in our experiments, namely sequential pattern mining, ranking, and active learning.

2.2.1 Sequential pattern mining

Sequential pattern mining is one of the most important topics in the area of frequent pattern mining, and in data mining in general (Han et al. 2007). The task of sequential pattern mining is the extraction of all frequent subsequences with the support³ greater than a specified threshold. Informally, a sequence a is a subsequence of a sequence b if one can remove items from sequence b (without changing their order), to finally get sequence a . Due to the importance of the task, various approaches have been proposed. One of them, SPADE (Zaki 2001) is based on breadth-first search and Apriori pruning on the vertical data format. This method, along with GSP, PrefixSpan, and SPAM, is considered one of the most popular approaches to discover sequential patterns (Fournier-Viger et al. 2017). For formal definitions and a detailed review see, e.g., Mabroukeh and Ezeife (2010).

2.2.2 Ranking

Learning to rank is a widely studied area of machine learning which was originally studied in the context of automatic ranking of web search results in the information retrieval community. However, it proved to be useful in many other areas, such as statistical machine translation (Watanabe 2012) or online advertising (Ciaramita et al. 2008). The task of learning to rank is the construction of a model which is able to sort new objects according to their degree of importance. The approaches for machine-learned ranking can be divided into three groups: the pointwise, the pairwise, and the listwise approaches. Pointwise methods make use of classification or regression techniques in order to predict a score for each object, which is subsequently used for sorting. Those methods entirely ignore the fact that objects are organized into groups and do not take into account any mutual dependency between the objects. An idea of predicting the order of each pair of objects is explored by pairwise methods. Those methods often apply adapted classification algorithms to decide if one element in the pair is superior to the other. They, similarly to pointwise approaches, do not take into account the position of elements in the final ranking. Finally, listwise approaches directly optimize metrics defined

³ The support of a subsequence is equal to the number of sequences that contain that subsequence in the database.

on a whole list of objects. A review of those methods was provided, for instance, by Liu (2009).

2.2.3 Active learning

Active learning is a subarea of machine learning which studies algorithms that are able to learn from only partially labelled data (Settles 2012). In such approaches, training examples are chosen by the algorithm itself with the ultimate goal of constructing an accurate model using as few examples as possible. To obtain labels for the selected examples, the algorithm usually queries a human expert or some kind of oracle to annotate them. The research on active learning is primarily focused on solving classification tasks; however, a quite limited number of works on active learning to rank was also done, see, e.g., Donmez and Carbonell (2008) or Long et al. (2015).

3 Proposed approach

Our machine learning formulation of the task of building a word-formation network for a presumably under-resourced language is as follows: given a set of lexemes for the language and derivational edges for a small subset of lexemes, predict derivational edges for the remaining ones.

We solve the task by choosing base lexemes for individual lexemes (or leaving them parentless) independently. We use supervised machine learning for learning how to choose the best base lexeme candidate. However, for such a prediction we use binary features previously generated from the set of lexemes in an unsupervised way. More specifically, our approach consists of the following steps:

1. extraction of lexeme features that correspond to string patterns frequently appearing in the lexeme set,
2. finding a set of most promising base lexeme candidates for each lexeme, ranking the candidates and choosing the one with the best score, or leaving the node parentless,
3. cycle elimination, which is needed since the greedy parent-selection approach does not guarantee the resulting structure to be free of cycles.

More details on the individual steps are presented in the following subsections and a general overview of our approach in the form of a diagram is depicted in Fig. 2.

3.1 Extracting features from the lexeme set

In the proposed experiments, lexemes are taken as sequences of alphabetic characters (we do not assume any other information about lexemes). Due to the subsequent supervised machine learning step, we need to turn lexemes into feature vectors. A generation of a reasonable set of features is needed not only because of the computational complexity of the ranker, but the feature set also strongly

Fig. 2 An overview of the proposed approach for the construction of word-formation networks

influences the overall generalization power of our approach. In other words, using, e.g., all substrings of all lexemes as binary features would lead not only to issues with computational tractability, but given the presumably very limited training data for selection of base lexemes, it would make the whole approach prone to overfitting.

We decided to use the sequential pattern mining technique for finding frequent subsequences. To perform this task we selected the SPADE algorithm (Zaki 2001) since it is computationally efficient for usual lexicon sizes and its implementation was readily available (however, any other algorithm that solves the sequential pattern mining problem could be adopted in a straightforward way too). The algorithm treats each word as a sequence of characters and the lexicon is interpreted as a database of words. Hence, the resulting subsequences are in fact lists of characters that often occur in a particular order. The examples of such frequent subsequences in the Polish lexicon are (n, i, e) , (o, w, y) and $(n, o, \acute{s}, \acute{c})$. Our hypothesis is that by finding subsequences covered by a lot of words, we will be able to discover useful morphological patterns in the lexeme set. This is the case for the aforementioned examples of frequent subsequences. Despite the fact that the subsequences have very general form, a Polish native speaker will easily recognize that the subsequence (n, i, e) could be transformed into *nie* ‘no’ (negation particle). Similarly, (o, w, y) and $(n, o, \acute{s}, \acute{c})$ could be later converted into *owy* ‘-al’ and *noś* ‘-ity’ (both are suffixes frequently used in Polish adjectives and nouns, respectively). Obtaining such meaningful features is the goal of further processing steps.

We represent each frequent subsequence as a regular expression. For example, the aforementioned subsequence (n, i, e) will be further denoted as *n*i*e*$ where $^$ and $*$ mark the beginning and the end of the word, respectively, and $*$ represents any string (including an empty one). At this point, one faces two problems: first, the extracted patterns are too general and second, the number of frequent subsequences is large. Hence, we proceed with a procedure whose goal is to make the patterns more specific but also more meaningful. A method for pruning the set of frequent subsequences is introduced below.

In order to make patterns more specific, we apply a greedy approach which iteratively tries to delete one of the symbols of any string (*) from the pattern. If the deletion of that symbol results in a small decrease in the support (less than a specified threshold), the newly created regular expression is accepted. The execution of this procedure results in more specific patterns with some of them having a linguistic interpretation. For example, $\wedge n * i * e * \$$ is replaced by $\wedge nie * \$$, which is a prefix used for the creation of negated forms in Polish, e.g., *dobry* ‘good’ and *niedobry* ‘bad’; $\wedge * o * \$ * \acute{c} * \$$ is converted to $\wedge * o \acute{s} \acute{c} \$$, which is a common suffix for Polish nouns, e.g., *męski* ‘manly’ and *męskość* ‘manhood’, *żwawo* ‘briskly’ and *żwawość* ‘briskness’. Since many other methods for automatic discovery of suffixes and prefixes have been proposed in the literature, it is important to note that our approach is able to detect more complex patterns than suffixation only. For example, patterns like $\wedge nie * o \acute{s} \acute{c} \$$, which is a common pattern of negated nouns derived from adjectives (e.g., *nieżyźność* ‘infertility’ from *żyźny* ‘fertile’, *niesakralność* ‘non-sacredness’ from *sakralny* ‘sacred’), or $\wedge * cz * o \acute{s} \acute{c} \$$ corresponding to nouns derived from adjectives containing *cz* digraph are also constructed. The latter pattern can be helpful in recognizing derivations where *k* was changed to *cz*, e.g., *lalka* ‘a doll’ to *lalczyność* ‘being-like-a-doll’ or *fizyczność* ‘physicalness’ from *fizyka* ‘physics’.

Next, one must deal with a high number of patterns generated. We have observed that there are groups of patterns matching almost the same set of lexemes. For example, $\wedge * cz * no \acute{s} * \$$ and $\wedge * cz * no \acute{s} \acute{c} \$$ have approximately the same support and cover the same lexemes, so keeping both of them seems to be redundant. In order to detect such pairs of redundant patterns, we perform a specific correlation analysis. First, we describe each lexeme in the lexicon by a binary feature vector. Each previously created regular expression is converted into one feature which takes 1 when the pattern matches the lexeme, and 0 otherwise. Having such representation of the lexicon, we are able to measure the association between patterns by calculating the phi coefficient (see e.g., Kotz et al. 2006). We consider a pair of patterns as a redundant one if its phi coefficient is greater than 95 %. Then, we shrink the number of patterns by selecting the most specific pattern from each indicated pair. This results in a considerably smaller set of patterns.

3.2 Base-lexeme candidate ranking and building the trees

The above-mentioned representation of lexemes as binary feature vectors facilitates using machine learning techniques for the construction of a word-formation network. Since the network is a forest composed of rooted trees, we can construct the network by merely finding a base word for each lexeme (or leaving the lexeme unattached). We do it for each lexeme independently; the only more global optimization operation is the deletion of cycles, as described below.

Once we decide to construct the network in such a greedy fashion, it might seem that we could use some binary classification technique simply for predicting the presence or absence of derivational edges within individual lexeme pairs. However, it would be very naive to assume that such a prediction would always deliver at most one base lexeme per lexeme. The most positive out of all positively predicted

parents would have to be chosen anyway. That is why we believe that it is more adequate to formalize the task as a ranking problem: we want to learn to find a scoring function that makes the correct parent the winner in the set of all plausible candidates.⁴ But even with this approach two additional subproblems are faced:

1. computing the score for each possible pair of lexemes implies quadratic time complexity, which would be quite computationally demanding given that a particular lexeme set can easily contain hundreds of thousands of lexemes (or even one million, as in the case of the Czech data); that is why we limit ourselves to computing the ranking score only for a relatively small set of most promising parent candidates for each lexeme, pre-selected by some other method,
2. derivational tree roots should remain parentless; in other words, we should be able to reliably recognize situations in which no parent is to be assigned.

We solve the former problem by restricting the list of possible base words to the 100 most similar words according to the Proxinet measure (Hathout 2009, 2014). Proxinet is a graph-based lexeme similarity measure which was designed for detecting morphological analogies. The measure is based on a comparison of two lexemes using all possible character n-grams. The higher the number of overlapping n-grams, the higher the value of lexemes' similarity. This measure also applies a specific graph weighting schema which results in rare n-grams contributing more to the overall similarity value. Below, we give examples for three lexemes:

- *niedobry* 'not good': *przedobry* 'very good', *nieprzedobry* 'not very good', *dobry* 'good', *arcydobry* 'excellent', *niedobrany* 'not well-matched', ...
- *męskość* 'manhood': *męsko* 'in a manly manner', *męski* 'manly', *niemęskość* 'non-manhood', *półmęskość* 'half-manhood', ...
- *harcerzyk* 'scout (diminutive)': *harcerz* 'scout', *harcerzować* 'to scout', *rycerzyk* 'knight (dimin.)', *pancerzyk* 'armour (dimin.)', *harcerstwo* 'scouting', ...

After constructing the candidate lists, a ranker can be trained to sort the lists in such a way that the correct base word gets the highest position/score. Even a relatively small training set provided by a linguist seems to be enough to train ranker parameters. Exemplary candidates are listed below together with their scores assigned by a ranker. The selected parent for each lexeme is underlined:

- *niedobry* 'not good': *dobry* 'good' 5.89, *przedobry* 'very good' 4.55, *arcydobry* 'excellent' 4.54, *niedobrany* 'not well-matched' 3.98, *nieprzedobry* 'not very good' 2.33, ...
- *męskość* 'manhood': *męski* 'manly' 8.93 *męsko* 'in a manly manner' 6.72, *niemęskość* 'non-manhood' 2.38, *półmęskość* 'half-manhood' 2.33, ...
- *harcerzyk* 'scout (diminutive)': *harcerz* 'scout' 10.34, *harcerzować* 'to scout' 6.78, *harcerstwo* 'scouting' 5.56, *rycerzyk* 'knight (dimin.)' 3.83, *pancerzyk* 'armour (dimin.)' 1.43, ...

The second problem is solved as follows: a derivational edge in the network is established when the difference between the rank of the first and second element on

⁴ In fact the winner takes it all and ordering in the tail of the list is more or less unimportant.

the sorted list exceeds a threshold provided by the user. Thus, if there is no clearly superior parent candidate for a lexeme, we consider the lexeme to be the root of the derivational tree.

3.2.1 Elimination of cycles

One disadvantage of constructing a network iteratively is the lack of any prevention from creating a cycle. In our approach, a network is constructed by finding a parent for each lexeme. This ensures that every node in a graph has at most one incoming edge. As a result, many of the graph's components will have the desired tree structure. Nevertheless, there is no guarantee that no cycle in the graph will be created (e.g., $A \rightarrow B \rightarrow C \rightarrow A$). To handle such a situation, we use a simple heuristic to eliminate cycles in the graphs. Our heuristic relies on the basic assumption that a derivative is usually longer than the base lexeme since, essentially, derivation consists of adding a suffix or a prefix to the base structure (cf. Sect. 1.2). Hence, we eliminate cycles by iterating over them and removing the first derivational edge between a shorter child and a longer parent. If all words in a cycle have the same length, we drop a randomly selected edge.

3.3 Active learning

It is important to note that an application of learning methods for the construction of a word-formation network unlocks many possibilities of reusing ideas already developed and studied by the machine learning community. One particularly attractive idea is the application of active learning methods.

The active learning algorithms allow for the iterative construction of machine learning models with the participation of human experts. Such approaches usually start learning with some small seed training set and interact with a user in order to augment the training data by potentially the most informative examples. The algorithm queries the user about the labels of selected data points, which minimizes the error rate as well as user workload.

Although the sizes of training sets considered in the current work are already fairly small, one can imagine a linguist providing an even smaller set of derivational pairs and contributing additional pairs requested by the algorithm, further limiting the human effort to create a word-formation network and/or improving its quality. We decided to verify this possibility by incorporating an active learning element into our machine learning approach.

As our aim is to rather present a kind of a feasibility study than looking for the most efficient solution, we decided to examine a relatively straightforward adaptation of selective sampling technique (Cohn et al. 1994) to our learning-to-rank method. In our active approach, we start off with a rather deficient dataset of derivational pairs and apply them to train our ranking system. Later, after following our usual network construction procedure (construction and ranking of candidate sets, cycle elimination etc.), we select lexemes for which the ranker had the lowest

confidence⁵ while predicting their base word. Then, we ask the linguist to assign base lexemes to the selected lexemes. Then we merge the user input with the initial training set. Finally, this procedure is repeated several times until reaching the required size of the training set or until exhausting the anticipated time to complete the task.

4 Data

We apply the introduced method to two Slavic languages, namely Czech and Polish, and two Romance languages, namely French and Spanish. For each of the languages, we employ the following types of datasets:

1. a set of lexemes,
2. a set of derivational edges for training the ranker,
3. a (disjoint) set of derivational edges for evaluation purposes.

A set of lexemes, the bigger the better, is needed for unsupervised extraction of frequent string patterns. For Czech and French, we adopt the underlying lexeme sets from their derivational resources (DeriNet 1.7 and Démonette 1.2), while for Polish and Spanish we had to use other data sources.

A relatively small good-quality set of derivational edges is used for a (supervised) training of a ranker that learns to choose a base lexeme for a given lexeme from possible candidates. For Czech and French, we use equal-sized subsets of the two existing derivational networks. For Polish and Spanish, we annotated derivational edges for a randomly selected subset of lexemes ourselves.

Another good-quality set of annotated derivational edges is used for estimating the achieved performance in the task of derivation prediction, in terms of precision and recall. If no existing gold-standard derivational data are available to us (which was the case of Polish and Spanish, again), we substitute evaluation of the generated derivations against some gold data by manual checking of a random sample selected from the predicted derivational links; we denote this evaluation method as *post-hoc check*.⁶

Unlike some other subfields of language data resources, the world of derivational morphology has not developed any popular data standards yet which could be compared, e.g., to the CoNLL data structure and file format, which is used in dependency treebanking for more than a decade and is applied to tens of languages within the Universal Dependencies project now (Nivre et al. 2016). That is why we had to gather the data from various sources and convert them into a unified form ourselves. The following subsections describe this data preprocessing for each

⁵ Since we are interested in the top position of the ranking only, we defined ranker confidence as the rank difference between the top two positions.

⁶ We are aware of the danger that mechanical search for disagreements between output of a method and previously annotated gold-standard data might—due to mostly psychological reasons—lead to a different result than evaluating the number of errors found by an annotator in the output of the method, even if the annotator person is the same. However, we don't expect this effect to be substantial in our study, as even the post-hoc checks were done very thoroughly.

language individually. For all languages, we used the data structure that is native for the Czech derivational network. Table 1 summarizes main quantitative properties of the assembled data collection.

Comprehensive linguistic summaries on the word-formation systems of all four languages involved in our experiments are provided by Müller et al. (2016), which was used as the primary reference resource in the research.

4.1 Czech

We used DeriNet version 1.7 in this study. DeriNet 1.7 contains 1,027,655 Czech lexemes and 808,682 derivations. The whole lexeme set was used for sequence pattern mining. The ranker was trained using a randomly selected sample of 1600 edges of DeriNet. The final evaluation was performed using the remaining edges in the network. Both samples are disjoint.

Czech is an exception among the four languages chosen for our study. Not only is DeriNet a large and good-quality data resource (mostly hand-annotated or hand-checked), but its derivational data model is identical with the model adopted in this article, and thus, the data was adopted as is, without any preprocessing.

4.2 French

We based our experiments with French on the Démonette derivational network (Hathout and Namer 2014). Like in the case of DeriNet for Czech, it served both as a source of lexeme set and as a source of reliable derivational annotation, used both for ranker development and for overall evaluation. The only difference is that the data from Démonette had to be arranged first, as Démonette itself is basically a collection of several different derivational resources, and that morphological clusters do not have the shape of a rooted tree, because between related lexemes there are derivational edges in both directions. Also, some indirect edges (resulting from a composition of two or more direct edges) are explicitly stored in the data. After pruning such relations and collapsing words with identical pairs of the lexeme string and part-of-speech category into single nodes, we received a forest composed of 22,570 nodes and 13,811 edges.

4.3 Polish

The lexeme base which we use for Polish comes from the Grammatical Dictionary of Polish (Saloni et al. 2017), which is a comprehensive lexicon of the Polish language, covering more than 261 thousand lexemes. This dictionary is quite popular in the Polish NLP community as it was used in the creation of many resources and tools such as Morfeusz morphological analyzer (Woliński 2006) and the Great Dictionary of Polish (Zmigrodzki 2011).

Using a random sample from this lexeme set, we annotated manually a small dataset consisting of 1554 pairs of base words with their derivatives. Polish native speakers, who created the training set, were asked to provide examples of as many different derivation patterns as possible. The construction of the training set took

Table 1 Data resources used for training and evaluation of our approach

Lang.	Set of lexemes		Training data		Evaluation data	
	Origin	#L	Origin	#D	Origin	#D
Czech	DeriNet 1.7	1,027,655	DeriNet 1.7	1600	DeriNet 1.7	801,804
French	Démonette 1.2	22,570	Démonette 1.2	1600	Démonette 1.2	12,211
Polish	Gramm.Dict.	261,819	Our gold. annot.	1554	Post-hoc check	400
Spanish	Leffe	153,829	Our gold. annot.	1026	Post-hoc check	400

#L stands for the number of lexemes. #D stands for the number of derivational edges

approximately 12 man-hours. This dataset was used for training and evaluation of the ranker. The overall quality of predicted derivations was evaluated using post-hoc checking.

4.4 Spanish

Our lexeme set for Spanish is based on that of the Leffe lexicon (Molinero et al. 2009). This lexicon in its original form did not fit our needs perfectly as it contained many foreign names (e.g., names of French villages), while at the same time some common words were missing in it, so we arranged it in two directions:

- we removed lexemes containing parenthesis, question marks, exclamation marks and punctuations signs and also digits,
- we merged this reduced set with lexemes from two other resources, namely:
 - the DRAE lexicon, currently posted to the public domain via GitHub,⁷
 - the Morales Flores lexicon.⁸

This resulted in a set containing 153,829 lexemes. We annotated manually a dataset containing 1026 randomly selected lexemes and their antecedents for training the ranker and evaluation. Post-hoc checking was used for the overall evaluation, as in the case of Polish.

5 Experiments and evaluation

The performance of the proposed approach is evaluated in several experiments on four selected, morphologically rich languages (Czech, French, Polish, and Spanish).

In these experiments, our main goals are the following:

- to analyze the performance of a machine-learned ranker trained on automatically generated features,

⁷ <https://github.com/olea/lemarios>.

⁸ http://www.cpimario.com/cm_util.html.

- to study the possibility of using previously trained ranker to construct a word-formation network,
- to investigate the potential of combining active learning methods with our method.

5.1 Experimental setup

The experiments are performed using the Python's version of XGBoost machine learning library (Chen and Guestrin 2016), which provides an efficient parallel implementation of Gradient Boosting Decision Trees (GBDT) algorithm. GBDT are quite complex learning systems whose base idea relies on boosting, a classic ensemble technique which trains component learners iteratively. The first component of an ensemble learns the training set as is and the task of the second component is to learn how to correct the errors made by the first one. The following learners are consequently trained to fix the errors of the already constructed components of the ensemble, which allows this learning technique to achieve high predictive performance in practice. Gradient boosting generalizes the idea of boosting to different loss functions, including the loss functions related to the problem of ranking. The GBDT algorithm was selected as a machine-learned ranker for our experiments because it is considered one of the state-of-the-art methods for learning to rank (Tyree et al. 2011; Ke et al. 2017). The GBDT ranking model implemented in XGBoost library belongs to the family of pairwise approaches and is simply based on minimization of the pairwise loss.

As for hyperparameter settings, besides choosing the optimization of the pairwise loss, we also set the number of decision trees to 100 with a maximum depth of 40—the values of remaining parameters were left on their default values.

Another essential part of our machine-learning based approach is the feature construction process driven by sequential pattern mining techniques. To perform this task, we used the Java-based SPMF data mining library, which includes open-source implementations of over a hundred pattern-mining methods (Fournier-Viger et al. 2016). As a sequential pattern mining algorithm, SPADE (Zaki 2001) was chosen due to its simplicity and the fact that it meets the requirements of our method, i.e., it extracts sequential patterns with the support above a given threshold. Other computational aspects such as time and memory consumption were not investigated in our experiments. The threshold on minimal pattern support was set to 1% in our experiment.

5.2 Ranker evaluation

In the first set of experiments, we evaluate the performance obtained by ranker models that use the features automatically generated by sequence pattern mining.

First, we apply the SPADE algorithm to each lexeme set which resulted in roughly 27 thousand frequent subsequences for Polish, almost 36 thousand for Czech and 10 thousand for each Spanish and French. Those numbers are quite substantial because from each pattern a feature is created and our aim is to create an

accurate model while keeping training set small. However, our filtering technique based on the phi coefficient substantially reduced the number of patterns for all considered languages. For instance, it shrinks the set of 27,551 patterns for Polish to only 13,441 frequent subsequences. By converting those subsequences to regular expressions (as described in Sect. 3), a feature vector for each training edge is created. Besides our automatically generated features, we add two additional and more general features: the length of the common prefix and the length of the common suffix.

From a list of derived lexemes and their base lexemes provided by linguists, we automatically constructed a training set for the ranker. In this place, we would like to point out that the training set for the ranking task consists of training groups rather than of independent training examples that are usual in classification and regression tasks. The group in the training set consists of data points with ranks assigned to them which indicate the correct order of elements in the group (from the highest rank to the lowest). This order should be learned by the model and potentially reconstructed during prediction, also for other unseen groups of elements. In our study, we have constructed training groups containing one lexeme together with 100 candidates for becoming its base lexeme. The candidates were selected by performing the nearest neighbor search as indicated in Sect. 3. In each group, the rank of the true base word is set to 1, whereas the rank of the remaining candidates is set to 0.

We evaluated our approach using fivefold cross-validation. For Polish we obtained the accuracy of 82.3% without applying any threshold on the confidence of the ranker. However, since we prefer precision to coverage, we have chosen a threshold which allowed us to obtain 98.8% of precision with the recall of 38.2%. Taking Spanish as another example, our method with properly selected threshold was able to get 98.1% precision with the recall of 27.7%. Such thresholded models were used for all the experiments.

To summarize, these results demonstrate the capability of the ranking model to learn derivation rules quite accurately, although providing only a limited recall.⁹

5.3 Evaluation of the semi-automatic construction of word-formation networks

Although the results presented in the previous section are encouraging, they do not directly prove the ability of the model to construct a whole word-formation network outside of the very limited lexemes in the training set. For instance, our training sets reflect the derivational pairs provided by a linguist. Hence, each lexeme in the data has a base word which is not quite true in the general setting. Yet another problem is the lack of a comparison with some other possible methods for network construction. We performed further experiments in order to clarify these issues.

At this point, we start differentiating between resource-rich (Czech, French) and resource-poor languages (Polish, Spanish). The performance of our method on

⁹ We have also tested the performance of a standard classification model instead of the learning-to-rank one, but we obtained inferior results.

languages which already have a developed derivation resource are evaluated against this resource. It means that the resource is treated as a gold standard: the derivations contained in the resource are assumed to be correct and the absent links are presumed to be non-existent. On the other hand, the assessment of resource-poor languages is performed manually by a language expert. The decision to process the considered languages differently was made in order to limit the amount of work done by linguists. Additionally, the already created resources are considerably bigger than those which could be provided by our evaluators.

The manual evaluation of the results for resource-poor languages was performed in two independent steps, separately for precision and recall. First, a random sample of 200 derivational edges was taken out of all automatically constructed edges. Then, a linguist checked each edge, and finally, precision was calculated as the ratio of correct edges to the total number of edges. In order to evaluate recall, a sample of 200 lexemes was selected randomly from the lexeme set. Next, the evaluator annotated each lexeme as a correct one either if it had correctly assigned a base word, or if it was correctly left parentless. The recall was computed as the number of correct lexemes divided by the total number.

Table 2 shows the precision and recall achieved by our approach together with the number of features generated automatically for each language. Apart from Czech, on every language our method obtained a high precision of around 95% and a considerable recall. One possible explanation for a lower precision on Czech is that using the Czech lexeme set results in the substantially higher number of features as compared to other languages; the growing dimensionality of the problem coupled with consistently keeping the training set small could cause induction of a less accurate model. While increasing the size of the training set could help in achieving better results, it is also true for the other languages. We avoided doing so because one of our goals was to infer word-formation networks from easily created training sets with a small linguist's workload and the precision of 91% we still consider a success. It is notable that our algorithm was able to create a network with 385 thousand derivational edges starting with only 1600 derivational pairs. It is even more remarkable for French, where our method achieved high precision and quite generous recall of 75% from a training set which could be easily created in one and a half day by a linguist.

The method also revealed the potential for the creation of word-formation networks for resource-less languages. The model trained on Spanish data was able

Table 2 The precision, recall and F-measure achieved by our method on the two resource-rich and two resource-poor languages as well as the number of automatically generated features and derivational edges

Language	Precision (%)	Recall (%)	F-measure (%)	No. of features	No. of edges
Czech	91.1	52.6	66.7	30,808	385,854
French	94.2	74.6	83.2	9012	9060
Polish	95.0	34.0	50.1	13,441	74,985
Spanish	94.9	44.0	60.1	9321	18,503

to discover 18.5 thousand derivational edges with the manually evaluated precision of 94.9% and recall of 44.0%.

The model trained for Polish lexemes was able to create more than 53.5 thousand links in the network. We estimated its precision and recall to 97% and 26.5%, respectively. Encouraged by the high precision of the constructed network, we decided to add the extracted derivational edges to our existing gold dataset for training. In this way, we obtained around 55 thousand lexeme pairs in the training set without any additional effort of language experts, although the training set has become slightly noisy. By applying our approach to this larger set, we have obtained almost 75 thousand derivational edges with 95% of precision and 34% of recall.

We also investigated the impact of the training set size (measured as the number of derivational pairs) on the quality of the constructed network. We repeated our experiment with a training set of 50 derivational pairs, 100 pairs, 200 pairs, etc. in each step doubling the size of training set until 1600 pairs were achieved. Figure 3 presents precision, recall and F-score in each iteration for French data, averaged over 10 randomly selected training sets. Results reveal that lowering the size of training data decreases the method's performance as expected. However, the loss of performance progresses rather slowly, e.g., 1.5% lower F-score can be obtained with half-sized training data. The plot also suggests that by the additional effort of language experts put into the construction of a bigger training set, the method can further improve its results.

Additionally, we compared the performance of our machine learning approach for word-formation network construction with an alternative approach based on the substitution rules provided by a human expert. We asked a language expert to provide us with 20 most productive and reliable derivational rules for each resource-less language. Each substitution rule was applied to every matching lexeme; however, the rule was creating a derivational edge in the network only when the

Fig. 3 Precision, recall, and F-score for different sizes of the training set for French data. Each measurement was repeated ten times; error bars represent standard deviations

Table 3 A comparison of precision and recall for machine learning and rule-based approach

Language	Method	Precision (%)	Recall (%)
Polish	Rules	90.0	66.7
	ML	95.0	34.0
Spanish	Rules	84.0	59.0
	ML	94.9	44.0

lexeme resulting from rule execution was already present in the lexeme set. This procedure allowed the linguist to create much more general and productive rules. The comparison of rule-based and machine learning approaches is presented in Table 3.

For both Polish and Spanish, the recalls obtained by the hand-crafted rules are considerably higher than those by the machine learning approach. However, the method based on machine learning is substantially better in terms of precision. Since the provided rules are the very productive ones, we would like to emphasize that further development of a higher number of hand-crafted rules will not increase the recall substantially. Furthermore, the impact of applying the rules in a particular order will grow with their number. This means that substitution rules will have to be designed with special caution and to be thoroughly tested. Conversely, one can expect the quality of the machine learning approach to improve together with the addition of more training examples. Also, a particular rule typically only handles one type of derivation while the machine learning approach is able to connect lexemes derived in plenty of different ways.

5.4 Evaluation of active learning method

In this subsection, we describe our experiment performed in order to investigate the potential of combining an active learning method, namely selective sampling, with our network construction method.

Figure 4 presents the results of our active learning method applied for the Polish lexeme set where we started with 1000 randomly selected derivational pairs from the training set used in the previous experiments. Six iterations of the active method were performed, and after each of them, a linguist was asked to label the 100 most uncertain lexemes. Recall that our Polish training set had almost 1600 derivational pairs; hence, in the last iteration, the active method used precisely as much training data as the previous static approach.

The results look quite favourable and show advantages of the active learning approach over the static approach, although the active learning approach requires notably higher computational power. After only four iterations, the approach caught up with the static method, creating more derivational edges with comparable precision. Therefore, an application of this method in the first place would have saved about an hour and a half of linguist's effort, while obtaining a similar result. The number of derivational edges constructed by the method further improved in

Fig. 4 The number of constructed derivational edges (the solid broken line) and their precision (the dashed broken line) with respect to the number of active learning iterations as applied to the Polish lexeme set. Horizontal lines depicts corresponding values for the static approach

consecutive iterations, and it seems to have had a good chance of continuing to grow had more iterations been performed. Finally, in the last iteration, the algorithm constructed almost seven thousand more edges (60,223) compared to the static approach, substantially improving recall (29.5%) and achieving a slightly higher precision (97.5%). The whole execution of this method requires a size of labelled data analogous to the previous one, hence, a comparable linguist's workload. It is also noteworthy that, as described in Sect. 5.3, the static model which we are comparing to was trained on a training set augmented by a previous model which would also be possible in this case. We believe that this result could be further improved by applying more elaborate solutions for active learning to rank.

6 Discussion

In order to better investigate the operation of the proposed method, we decided to perform an exploratory study of the created word-formation networks. To this purpose, we have used a search engine named DeriSearch (Vidra and Žabokrtský 2017), which provides a useful visualization tool for word-formation networks. In the following paragraphs, we discuss some general patterns found in our study by using several examples and providing several visualizations of parts of our word-formation networks. In our discussion, we focus on both resource-less languages, namely Polish and Spanish as we find it more interesting than presenting reconstructed parts of already existing resources.

An example of a derivational tree of a reasonable size from the Polish word-formation network is presented in Fig. 5. The word *Szczecin* in this figure is an

Fig. 5 The noun *Szczecin* (the capital of West Pomeranian Voivodeship) and derivationally related lexemes (displayed as a tree structure) in the Polish word-formation network

Fig. 6 The noun *herbata* ‘tea’ and derivationally related lexemes (displayed as a tree structure) in the Polish word-formation network

example of a proper noun (a large Polish city) which is a base word for the derived adjective (*szczeciński* ‘related to Szczecin’), the noun being the name of its inhabitants (*szczecinianin*) and even for the diminutive form which happens to be the name of another smaller city (*Szczecinek*). All these lexemes are correctly connected in the network, despite the fact that, e.g., the word-formation process of *szczeciński* is potentially difficult because of the occurring alternation of *n* to *ń*. The figure shows that the tree has multiple layers with further derivatives, e.g., *podszczeciński* ‘related to Szczecin’s surroundings’.

Another derivational tree for *herbata* ‘tea’ which is depicted in Fig. 6 also seems correct and complete, containing derived adjectives (e.g., *herbaciarny*), nouns (e.g., *herbaciarnia* ‘tea shop’), negated forms (e.g., *nieherbaciarny*) and diminutives (*herbatka*). One can notice the lack of adverbs; however, they are absent in our lexeme set for this particular case. It is apparent that these trees are not the biggest ones in our network since the trees for the most productive words like *być* ‘to be’ or *robić* ‘to do’ may have several dozen nodes.

As for the Spanish network, many trees seem to be faultless. For instance, in Fig. 7 the derivational tree for the verb *ilustrar* ‘to illustrate’ is presented. Similarly to the visualizations of the Polish network, one can see that this lexeme is correctly linked to many derivatives. The derived adjectives like *ilustrativo* ‘illustrative’ or

Fig. 7 The verb *ilustrar* ‘to illustrate’ and derivationally related lexemes (displayed as a tree structure) in the Spanish word-formation network

Fig. 8 The verb *nauczyć* ‘to learn.PERFECTIVE’ and derivationally related lexemes (displayed as a tree structure) in the Polish word-formation network

ilustrado ‘illustrated’ as well as derived nouns (e.g., *ilustración* ‘illustration’) are properly connected in the network.

One can also see that the presented approach can also handle some untypical cases. For example, *ponauczać* ‘to teach.PERFECTIVE’ is a Polish verb that does not have an imperfective form. Even though a blind application of morphological rules would result with *ponauczyć* ‘to teach’ (e.g., *nauczyć* and *nauczać* are correctly connected), this verb is correctly connected with another perfective form *nauczać* ‘to teach’, as can be seen in Fig. 8.

However, one can find some errors in the visualized network structure. For example, in Figs. 9 and 10 we present the derivational trees for *karta* ‘sheet’ and *karty* ‘card game’, respectively. First of all, we would rather expect these two trees to be mutually connected. The lack of the derivational edge between *karta* and *karty* causes the absence of a whole derivational subtree in the derivations of *karta*. Second, the words *Djakarta/Dżakarta* ‘Jakarta’ are definitely not derived from *karta* and *gokartowość* ‘go-cart’ is not derived from *kartowość*. Such errors related to the words adopted from other languages are quite frequent. Another example of such error is shown in Fig. 11, in which *pastoforium* (coming from Greek) should not be derived.

An evident problem of the Spanish network is a rather low recall, resulting in many lexemes having too small derivational trees. A good example of this is the derivational tree for the verb *hacer* ‘to do’ which is presented in Fig. 12. In our network, only the noun *hacedor* ‘maker’ is connected with this verb and the absence of many other lexemes is rather evident, e.g., *rehacer* ‘to redo’ or *deshacer* ‘to undo’ are lacking.

Another problem that is sometimes visible in tree visualizations and which is not reflected by precision/recall values is the impact of a single derivational edge on the overall correctness of a tree. For instance, the lack of a single derivational edge can separate a tree into two parts of considerable sizes making the original tree very incomplete. Similarly, the construction of an incorrect edge between two sizeable trees can result in a huge tree without a clearly defined morphological family.

Fig. 9 The noun *karta* ‘sheet’ and derivationally related lexemes (displayed as a tree structure) in the Polish word-formation network

Fig. 10 The noun *karty* ‘card game’ and derivationally related lexemes (displayed as a tree structure) in the Polish word-formation network

Fig. 11 The noun *pastor* ‘pastor’ and derivationally related lexemes (displayed as a tree structure) in the Polish word-formation network

Fig. 12 The verb *hacer* ‘to do’ and derivationally related lexemes (displayed as a tree structure) in the Spanish word-formation network

Comparing the network constructed by our machine learning approach with resources generated by hand-crafted rules, we observe that the semi-automatic method sometimes lacks in being consistent which is an inherent property of substitution rules. For instance, our approach has no problem in constructing the negated forms of adjectives but still we found that there is a quite considerable amount of adjectives for which the ranker was not confident enough to construct a derivational edge to its negated form. On the other hand, our machine learning approach is able to detect pairs of lexemes which are derived in a fairly complex

way, which would require the development of quite specific rules. Moreover, as already mentioned in Sect. 5, the development of substitution rules is not as easy as augmenting the training set for a learning algorithm.

Although adding derivational pairs is simple, one must realize that all the design choices of the word-formation network (see Sect. 1.2) have to be indirectly provided to the model by the training set. A language expert who creates it must be aware of it and stick to accepted design decisions. However, machine learning models tend to be quite robust with regard to a small portion of noise in the training data, which makes them fairly distinct from hand-crafted substitution rules.

7 Conclusion and future research

In the present article, a new semi-automatic approach for the construction of word-formation networks is presented. In particular, it applies sequential pattern mining to construct useful morphological features. To the best of our knowledge, no one has used these techniques in that context so far.

Our approach was successfully evaluated on two Slavic and two Romance languages, namely Polish, Czech, Spanish, and French. For all languages under study, the newly proposed method discovered plenty of valid derivational edges between base words and their derivatives. The resources created by the method are characterized by high precision, and their creation does not require large human effort.

Moreover, the networks induced for all four languages using the semi-supervised setting described in Sect. 5.3 have been published online, together with the sets of features extracted according to Sect. 3.1, and with our hand-annotated datasets mentioned in Sects. 4.3 and 4.4; in addition, the derivational edges generated for Polish have been merged with derivational edges from Polish WordNet, which resulted in the biggest resource modeling Polish derivations called Polish word-formation network.¹⁰

Naturally, the coverage of the created resources still could be improved. We have shown that our method can achieve substantially better results by leveraging the active learning framework without increasing the human effort. In such an approach, training examples which should be annotated are chosen by the algorithm itself with the ultimate goal of constructing an accurate model using as few examples as possible. We demonstrated that even the application of a simple selective sampling strategy leads to an improved quality of the network while keeping the size of training data constant. As a future research, the already discussed active learning framework could be studied more extensively. We hope that further investigation of more advanced approaches, such as Long et al. (2015) and Donmez and Carbonell (2008), would further improve the results. Moreover, some specialized approaches for improving the recall of derivational lexicons can be applied; e.g., the graph-theoretical approach of Papay et al. (2017).

¹⁰ All generated networks as well as our sets of features and hand-annotated datasets are available at <http://ufal.mff.cuni.cz/derinet>.

The possibility of using a cross-lingual transfer to the fully automatic construction of word-formation networks could also be explored. As mentioned above, derivational resources have been already developed for several languages, which makes it possible to employ those resources to the construction of word-formation networks for related languages. For instance, the Czech DeriNet may be used to extend the proposed Polish network. Each derivational edge from the Czech network could be translated into Polish using one of the available dictionaries, such as Treq (Škrabal and Martin 2017). Then, some notion of morphological similarity should be adopted in order to eliminate incorrect derivational edges. In the context of the present work, even the constructed candidate sets may be used to check if the indicated base word is morphologically related. The preliminary experiments which we have run so far are quite promising. The described approach was able to find over 40 thousand derivational edges but with rather poor precision. Roughly 40% of the discovered edges were correct; however, one-third of errors could be simply fixed by reversing the direction of the relation. We believe that there are many possibilities to improve this result, for instance by taking the translation probability into account.

Another issue of the current approach is that it is limited to derivation only. Hence, there is a need for developing novel methods for automatic discovery of other word-formation processes, particularly compounding. Additionally, some effort to design the representation of such enhanced resources is also needed.

A separate line of future research lies in applying word-formation networks to practical NLP tasks. The use of word-formation networks can allow the transfer of semantic similarities from rarer to more common lexemes within a language, for instance through the construction of more appropriate word embeddings. An experimental study in the recent work of Finley et al. (2017) on analogy completion through so-called word embedding arithmetic showed that derivational analogies between lexemes are modeled less accurately by classic word embedding methods than analogies based on, e.g., inflectional morphology. This indicates an opportunity for further enhancements of those methods, which could leverage this issue by the use of word-formation networks during the construction of word embeddings.

Another possible application of such language resources lies in machine translation systems. As described by Ševčíková and Žabokrtský (2014), some machine translation systems look for a translation equivalent with a particular part-of-speech category. Considering English to Czech translation, an adjectival attribute of an English gerund is often translated as a Czech adverb. Such adverbs for rare adjectives are often lacking in translation systems due to the scarcity of parallel corpora but could be easily found in a word-formation network with high coverage.

Finally, word-formation networks which model derivations in a linguistically adequate way may boost linguistic research in morphology (more specifically, e.g., on productivity in word-formation).

Acknowledgements This work was supported by the Grant No. GA19-14534S of the Czech Science Foundation. It has been using language resources developed, stored, and distributed by the LINDAT/CLARIAH-CZ project of the Ministry of Education, Youth and Sports of the Czech Republic (project LM2018101). Mateusz Lango also acknowledges the support from the Institute of Computing Science Statutory Funds and from the internship program of the Institute of Formal and Applied Linguistics,

Faculty of Mathematics and Physics, Charles University. The authors thank two anonymous reviewers for useful suggestions. They would also like to express their gratitude to Jonáš Vidra for making accessible the visualization software and to Ebrahim Ansari, Eva Hajičová, Jarmila Panevová, Jonáš Vidra, Anna Nedoluzhko, and Daniel Zeman for numerous valuable comments on the manuscript.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Apresjan, J., Boguslavsky, I., Iomdin, B., Iomdin, L., Sannikov, A., & Sizov V. (2006). A syntactically and semantically tagged corpus of Russian: state of the art and prospects. In *Proceedings of the 5th international conference on language resources and evaluation (LREC 2006)*, pp. 1378–1381.
- Baayen, H., Piepenbrock, R., & Gulikers, L. (1996). *CELEX 2. CD-ROM, linguistic data consortium*, LDC Catalog No.: LDC96L14, Philadelphia.
- Balvet, A., Barque, L., & Marín, R. (2010). Building a lexicon of French deverbal nouns from a semantically annotated corpus. In *Proceedings of the 7th international conference on language resources and evaluation (LREC 2010)*, pp. 1408–1413.
- Baranes, M., & Sagot, B. (2014). A language-independent approach to extracting derivational relations from an inflectional lexicon. In *Proceedings of the 9th international conference on language resources and evaluation (LREC 2014)*, pp. 2793–2799.
- Bonami, O., & Strnadová, J. (2018). Paradigm structure and predictability in derivational morphology. *Morphology*. <https://doi.org/10.1007/s11525-018-9322-6>.
- Chen, T., & Guestrin, C. (2016). XGBoost: a scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining (KDD 2016)*, pp. 785–794.
- Ciaramita, M., Murdock, V., & Plachouras, V. (2008). Online learning from click data for sponsored search. In *Proceedings of the 17th international conference on World Wide Web, ACM (WWW 2008)*, pp. 227–236.
- Cohn, D., Atlas, L., & Ladner, R. (1994). Improving generalization with active learning. *Machine Learning*, 15(2), 201–221.
- Cotterell, R., Kirov, C., Sylak-Glassman, J., Walther, G., Vylomova, E., McCarthy, AD., Kann, K., Mielke, S., Nicolai, G., & Silfverberg, M., et al. (2018). The CoNLL–SIGMORPHON 2018 shared task. In *Proceedings of the CoNLL SIGMORPHON 2018 Shared Task: Universal Morphological Reinflection*, pp. 1–27.
- Cotterell, R., Kirov, C., Sylak-Glassman, J., Walther, G., Vylomova, E., Xia, P., Faruqi, M., Kübler, S., Yarowsky, D., Eisner, J., & Hulden, M. (2017). CoNLL–SIGMORPHON 2017 shared task: Universal morphological reinflection in 52 languages. In *Proceedings of the CoNLL SIGMORPHON 2017 shared task: Universal morphological reinflection, association for computational linguistics*, pp. 1–30.
- de Paiva, V., Real, L., Rademaker, A., & de Melo, G. (2014). Nomlex-PT: A lexicon of Portuguese nominalizations. In *Proceedings of the 9th international conference on language resources and evaluation (LREC 2014)*, pp. 2851–2858.
- Dokulil, M. (1962). *Tvoření slov v češtině I: Teorie odvozování slov*. Prague: Academia.
- Donmez, P., & Carbonell, JG. (2008). Optimizing estimated loss reduction for active sampling in rank learning. In *Proceedings of the 25th international conference on machine learning, ACM (ICML 2008)*, pp. 248–255.

- Fellbaum, C., Osherson, A., & Clark, P. E. (2007). Putting Semantics into WordNet's "Morphosemantic" Links. In *Language and technology conference: Human language technology. Challenges of the information*, pp. 350–358.
- Finley, G., Farmer, S., & Pakhomov, S. (2017). What analogies reveal about word vectors and their compositionality. In *Proceedings of the 6th joint conference on lexical and computational semantics (*SEM 2017)*, pp. 1–11.
- Fournier-Viger, P., Lin, C., Gomariz, A., Gueniche, T., Soltani, A., Deng, Z., & Lam, H. T. (2016). The SPMF open-source data mining library version 2. In *Proceedings of the 19th European conference on principles of data mining and knowledge discovery (PKDD 2016) Part III*.
- Fournier-Viger, P., Lin, J. C. W., Kiran, R. U., & Koh, Y. S. (2017). A survey of sequential pattern mining. *Data Science and Pattern Recognition*, 1(1), 54–77.
- Habash, N., & Dorr, B. (2003). A categorial variation database for English. In *Proceedings of HLT-NAACL*, pp. 17–23.
- Hajič, J., Panevová, J., Hajičová, E., Sgall, P., Pajas, P., Štěpánek, J., Havelka, J., Mikulová, M., Žabokrtský, Z., & Ševčíková-Razímová, M. (2006). Prague Dependency Treebank 2.0. CD-ROM, linguistic data consortium, LDC Catalog No.: LDC2006T01, Philadelphia
- Han, J., Cheng, H., Xin, D., & Yan, X. (2007). Frequent pattern mining: Current status and future directions. *Data Mining and Knowledge Discovery*, 15(1), 55–86.
- Hathout, N. (2005). Exploiter la structure analogique du lexique construit: Une approche computationnelle. *Cahiers de lexicologie: Revue internationale de lexicologie et lexicographie*, 87, 5–28.
- Hathout, N. (2009). Acquisition of morphological families and derivational series from a machine readable dictionary. In: Montermini F, Boyé G, Tseng J (eds). *Selected proceedings of the 6th Décembrettes, Cascadilla proceedings project*, pp. 166–180.
- Hathout, N. (2010). *Morphonette: A morphological network of French*. CoRR, [arXiv:1005.3902](https://arxiv.org/abs/1005.3902).
- Hathout, N. (2014). Phonotactics in morphological similarity metrics. *Language Sciences*, 46, 71–83.
- Hathout, N., & Namer, F. (2014). Démonette, a French derivational morpho-semantic network. *Linguistic Issues in Language Technology*, 11(5), 125–168.
- Hathout, N., Namer, F., & Dal, G. (2002). *An experimental constructional database: The MorTAL project. Many morphologies* (pp. 178–209). Cascadilla: Cambridge Mass.
- Iacobini, C. (2000). Base and direction of derivation. In G. Booij, C. Lehmann, J. Mugdan, W. Kesselheim, & S. Skopeteas (Eds.), *Morphology. An international handbook on inflection and word-formation* (Vol. 1, pp. 865–876). Berlin: Walter de Gruyter.
- Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., & Liu, T. Y. (2017). Lightgbm: a highly efficient gradient boosting decision tree. In *Advances in neural information processing systems*, pp. 3146–3154.
- Körtvélyessy, L. (2016). Word-formation in Slavic languages. *Poznań studies in contemporary linguistics*, 52, 455–501.
- Körtvélyessy, L., Štekauer, P., Genči, J., & Zimmermann, J. (2018). Word-formation in European languages. *Word Structure*, 11, 313–358.
- Kotz, S., Read, C. B., Balakrishnan, N., & Vidakovic, B. (2006). *Encyclopedia of statistical sciences* (2nd ed.). Hoboken: Wiley.
- Kyjánek, L. (2018). Morphological resources of derivational word-formation relations. Tech. Rep. TR-2018-61, UFAL MFF UK, Prague.
- Lieber, R., & Štekauer, P. (2014). *The Oxford handbook of derivational morphology*. Oxford: Oxford University Press.
- Litta, E., Passarotti, M., & Culy, C. (2016). *Formatio formosa est. Building a word formation based Lexicon for Latin*. In *Proceedings of the 3rd Italian conference on computational linguistics (CLiC-it 2016)*, pp. 185–189.
- Liu, T. Y. (2009). Learning to rank for information retrieval. *Foundations and Trends in Information Retrieval*, 3(3), 225–331.
- Long, B., Bian, J., Chapelle, O., Zhang, Y., Inagaki, Y., & Chang, Y. (2015). Active learning for ranking through expected loss optimization. *IEEE Transactions on Knowledge and Data Engineering*, 27(5), 1180–1191.
- Mabroukeh, N. R., & Ezeife, C. I. (2010). A taxonomy of sequential pattern mining algorithms. *ACM Computing Surveys (CSUR)*, 43(1), 3.

- Molinero, M. A., Sagot, B., & Nicolas, L. (2009). A morphological and syntactic wide-coverage lexicon for Spanish The Leffe. In *Proceedings of the 7th conference on recent advances in natural language processing (RANLP 2009)*. Borovets, Bulgaria.
- Müller, P. O., Ohnheiser, I., Olsen, S., & Rainer, F. (2016). *Word-formation. An international handbook of the languages of Europe* (Vol. 4). Berlin: Mouton de Gruyter.
- Nivre, J., De Marneffe, MC., Ginter, F., Goldberg, Y., Hajič, J., Manning, CD., McDonald, R., Petrov, S., Pyysalo, S., Silveira, N., Tsarfaty, R., & Zeman, D. (2016). Universal dependencies v1: A multilingual treebank collection. In *Proceedings of the 10th international conference on language resources and evaluation (LREC 2016)*, pp. 1659–1666.
- Pala, K., & Hlaváčková, D. (2007). Derivational Relations in Czech WordNet. *Proceedings of the Workshop on Balto-Slavonic Natural Language Processing, 2007*, 75–81.
- Pala, K., & Šmerk, P. (2015). Derivancze—derivational analyzer of Czech. *Text, speech, and dialogue, lecture notes in computer science* (Vol. 9302, pp. 515–523). Berlin: Springer International Publishing.
- Papay, S., Lapesa, G., & Padó, S. (2017). Evaluating and improving a derivational lexicon with graph-theoretical methods. In *Proceedings of the first international workshop on resources and tools for derivational morphology*, pp. 73–82.
- Saloni, Z., Gruszczyński, W., Woliński, M., Włosz, R., & Skowrńska, D. (2017). *Słownik gramatyczny języka polskiego*. Olsztyn: University of Warmia and Mazury.
- Settles, B. (2012). Active learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 6 (1), 1–114.
- Ševčíková, M. & Žabokrtský, Z. (2014). Word-formation network for Czech. In *Proceedings of the 9th international conference on language resources and evaluation (LREC 2014)*, pp. 1087–1093.
- Shafaei, E., Frassinelli, D., Lapesa, G., & Padó, S. (2017). DERivCELEX: Development and evaluation of a German derivational morphology lexicon based on CELEX. In *Proceedings of the first international workshop on resources and tools for derivational morphology*, pp. 117–127.
- Škrabal, M., & Martin, V. (2017). Databáze překladových ekvivalentů Treq. *Časopis pro moderní filologii*, 99(2), 245–260.
- Šnajder, J. (2014). DerivBase.hr: A high-coverage derivational morphology resource for Croatian. In *Proceedings of the 9th international conference on language resources and evaluation (LREC 2014)*, pp. 3371–3377.
- Šojat, K., Srebačić, M., Pavelić, T., & Tadić, M. (2014). CroDeriV: a new resource for processing Croatian morphology. In *Proceedings of the 9th international conference on language resources and evaluation (LREC 2014)*, pp. 3366–3370.
- Štekauer, P., Valera, S., & Körtvélyessy, L. (2012). *Word-formation in the world's languages*. Cambridge: Cambridge University Press.
- Talamo, L., Celata, C., & Bertinetto, P. M. (2017). DerIvaTario: An annotated lexicon of Italian derivatives. *Word Structure*, 9(1), 72–102.
- Tyree, S., Weinberger, KQ., Agrawal, K., & Paykin, J. (2011). Parallel boosted regression trees for web search ranking. In *Proceedings of the 20th international conference on World Wide Web, ACM*, pp. 387–396.
- Vidra, J., Žabokrtský, Z. (2017). Online software components for accessing derivational networks. In *Proceedings of the first international workshop on resources and tools for derivational morphology*, pp. 129–139.
- Watanabe, T. (2012). Optimized online rank learning for machine translation. In *Proceedings of the 2012 conference of the North American chapter of the association for computational linguistics*, pp. 253–262.
- Woliński, M. (2006). Morfeusz—a practical tool for the morphological analysis of Polish. In S. T. Wierchoń, K. Trojanowski, & M. A. Kłopotek (Eds.), *Intelligent information processing and web mining* (pp. 511–520). Berlin: Springer.
- Žabokrtský, Z., Ševčíková, M., Straka, M., Vidra, J., & Limburská, A. (2016). Merging data resources for inflectional and derivational morphology in Czech. In *Proceedings of the 10th international conference on language resources and evaluation (LREC 2016)*, pp. 1307–1314.
- Zaki, M. J. (2001). SPADE: An efficient algorithm for mining frequent sequences. *Machine Learning*, 42 (1), 31–60.

- Zeller, B., Šnajder, J., & Padó, S. (2013) DErivBase: Inducing and evaluating a derivational morphology resource for German. In *Proceedings of the 51st annual meeting of the association for computational linguistics*, Vol. 1. Long Papers. pp. 1201–1211.
- Zmigrodzki, P. (2011). Polish academy of sciences great dictionary of Polish: History, presence, prospects. *Studies in Polish Linguistics*, 6(1), 7–26.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.