Derivational Morphological Relations in Word Embeddings

Tomáš Musil and **Jonáš Vidra** and **David Mareček**
Charles University, Faculty of Mathematics and Physics
Institute of Formal and Applied Linguistics
Malostranské náměstí 25, 118 00 Prague, Czech Republic
{musil,vidra,marecek}@ufal.mff.cuni.cz

Abstract

Derivation is a type of a word-formation process which creates new words from existing ones by adding, changing or deleting affixes. In this paper, we explore the potential of word embeddings to identify properties of word derivations in the morphologically rich Czech language. We extract derivational relations between pairs of words from DeriNet, a Czech lexical network, which organizes almost one million Czech lemmata into derivational trees. For each such pair, we compute the difference of the embeddings of the two words, and perform unsupervised clustering of the resulting vectors. Our results show that these clusters largely match manually annotated semantic categories of the derivational relations (e.g. the relation 'bake–baker' belongs to category 'actor', and a correct clustering puts it into the same cluster as 'govern–governor').

1 Introduction

Word embeddings are a way of representing discrete words in a continuous space. Embeddings are used in neural networks trained for various tasks, e.g. in neural machine translation (NMT), or can be pre-trained in various versions of language models to be used as continuous representations of words for other tasks. One of the most popular frameworks for training word embeddings is word2vec (Mikolov et al., 2013).

In this paper, we examine whether the word embeddings (trained on the whole words, not using any subword units or individual characters) capture derivational relations. We do this to better understand what different neural networks represent about words and to provide a base for further development of derivational networks.

Derivation is a type of word-formation process which creates new words from existing ones by adding, changing or deleting affixes. For example, the word "collide" can be used as a base for deriving e.g. the words "collider" or "collision". The derived word "collision" can be, in turn, used as a base for "collisional".

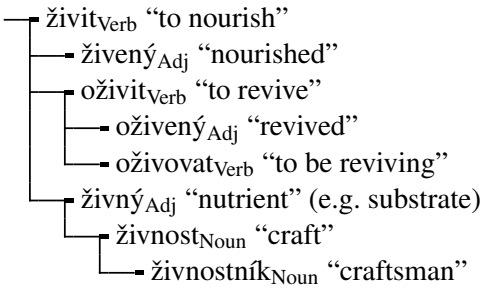

Figure 1: An excerpt from a derivational family rooted in the word "živit" (to nourish, to feed). Note that the word "oživený" (revived, rejuvenated), which can be derived from either "oživit" (to revive) or "živený" (nourished, fed), is arbitrarily connected only to the former, in order to simplify the derivational family to a rooted tree.

Words derived from a single root create derivational families, which can be approximated by directed acyclic graphs or (with some loss of information) trees; see Figure 1 for an example.

Derivational relations have two sides: form-based and semantic. For a pair of words to be considered derivationally related, the two words must be related both by their phonological or orthographical forms and by their meaning.

2 Related work

We have not found any prior work aimed specifically at derivational relations in word embeddings.

Cotterell and Schütze (2018) present a model of the semantics and structure of derivationally complex words. Our work differs in that we are examining how are derivational relations represented in preexisting applications.

Proceedings of the Second BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP, pages 173–180
Florence, Italy, August 1, 2019.

Gladkova et al. (2016) detect morphological and semantic relations (including some derivational relations) with word embeddings. Their approach is analogy-based and they conclude that their "experiments show that derivational and lexicographic relations remain a major challenge".

Gábor et al. (2017) explore vector spaces for semantic relations, using unsupervised clustering. They evaluate the clustering on 9 semantic relation classes. Our approach is similar, but we focus on derivational relations.

Soricut and Och (2015) use word embeddings to induce morphological segmentation in an unsupervised manner. Some of the relations between words that this approach implicitly uses are derivational.

3 Data

In this section, we describe the network of derivational relations and the corpora used in our experiments.

3.1 DeriNet

There are several large networks of derivational relations available for use in research, e.g. CELEX for Dutch, English and German (Baayen et al., 1995), Démonette for French (Hathout and Namer, 2014), DeriNet for Czech (Ševčíková and Žabokrtský, 2014) or DErivBase for German (Zeller et al., 2014). A more complete listing was published by Kyjánek (2018).

For our research, we chose to use the DeriNet-1.6 network mainly due to its large size – with over a million lemmata (citation forms), it is over three times larger than the second largest resource listed by Kyjánek (2018), DErivBase with 280,336 lemmata. Also, the authors are native speakers of Czech, which was necessary for the annotation of derivation classes (see Section 4 below). Large corpora are available for Czech (Bojar et al., 2016; Hnátková et al., 2014), which we need for training the word embeddings.

DeriNet is a network which approximates derivational families using trees – the lemmata it contains are annotated with a single derivational parent or nothing in case the word is either not derived or a parent has not been assigned yet. It contains 1,025,095 lemmata connected by 803,404 relations.

There is a fine line between derivation and inflection and in general, these processes are hard to separate from each other (see e.g. ten Hacken, 2014). Both change base words using affixes, but they differ in the type of the outcome: derivation creates new words, inflection only creates forms of the base word. DeriNet differentiates derivation from inflection the same way the Czech morphological tool MorphoDiTa (Straková et al., 2014) does – it considers the processes handled by the MorphoDiTa tool to be inflectional and other affixations derivational. This is in line with the Czech linguistic tradition (Dokulil et al., 1986), except perhaps for the handling of the two main borderline cases, whose categorization varies: negation (considered inflectional by us) and verbal aspect changes (considered derivational).

3.2 Word Embeddings

In our experiments, we compare the word embeddings obtained by the standard word2vec skip-gram model (Mikolov et al., 2013) with word embeddings learned when training three different neural machine translation (NMT) models (Sutskever et al., 2014; Bahdanau et al., 2015; Vaswani et al., 2017). The size of word embeddings is 512 for all the models.

NMT models are trained between English and Czech in both directions. We use the CzEng 1.6 parallel corpus (Bojar et al., 2016), section *c-fiction* (78 million tokens) and the Neural Monkey toolkit (Helcl et al., 2018)[1] for training the models. We experiment with three architectures:

- *RNN*: a simple recurrent neural (RNN) architecture (Sutskever et al., 2014) without attention mechanism, LSTM size 1,024
- *RNN+a*: RNN architecture with attention mechanism (Bahdanau et al., 2015), and
- *Transf.*: the Transformer (big) architecture (Vaswani et al., 2017) with 6 layers, hidden size 4,096 and 16 attention heads.

Unlike the standard setting in which embeddings of the source and the target words are shared in a common vector space, we use two separated dictionaries (each containing 25,000 word forms). We also do not use any kind of sub-word units. By this setting, we assure that the word vectors are not influenced by any other words that do not belong to the examined language. We extract the encoder word-embeddings from

[1] https://github.com/ufal/neuralmonkey

Czech-English NMT model and the decoder word-embeddings from the English-Czech model.

The word2vec system is trained on the Czech National Corpus (Hnátková et al., 2014), version *syn 4*, which has 4.6 billion tokens. It is a common practice (Mikolov et al., 2013) to normalize the resulting vectors, so that the length of each vector is equal to 1. We report results for both normalized and non-normalized vectors. In order to compare word2vec model with NMT models, we also train word2vec on the Czech part of the data used for training the NMT models.

All the word embeddings are trained on the word forms. To assign an embedding to the lemma from DeriNet, we simply select the embedding of the word form which is the same as the given lemma.

4 Annotation of Derivational Relations

The derivational relations in DeriNet are not labelled in any way. In this section, we describe a simple method of automatic division of relations into *derivation types* according to changes in prefixes and suffixes and then manual merging of these types into *derivation classes*.

When assigning a derivation type to a relation, we first identify the longest common substring of the two related words. For instance, for the relation "padat → padnout", the longest common substring is "pad". Then, we identify prefixes and suffixes using the '+' sign for addition and '-' sign for deletion. A sign after the string indicates a prefix and a sign before the string indicates a suffix. Our example "padat → padnout" would therefore belong to the derivation type "-at +nout", which means deleting the suffix "at" and adding the suffix "nout". Derivation type "na+" means to add the prefix "na", etc.

When applied on the DeriNet relations, we identified 5,371 derivation types in total. We selected only 71 most frequent types (only those that have at least 250 instances in DeriNet).[2] After that, two annotators[3] manually merged the 71 derivation types into 21 classes. The classes of derivations are listed in Table 1. The class *super+* contains derivations from nouns to nouns and from adjectives to adjectives. Except for insignificant noise in the data, each of the rest of the classes contain only derivations for one POS pair.

The classes were designed in a way to separate different meanings of derivations where possible, and keep different types with the same meaning together (e.g. '+ová' and '-a +ová', which derive feminine surnames).

5 Unsupervised Clustering

We want to know whether and how the derivational relations are captured in the embedding space. We hypothetize that in that case *the differences between embedding vectors* for the words in a derivational relation would cluster according to the classes we defined.

We perform unsupervised clustering of such differences using three algorithms:

- **kmeans**: K-means algorithm (MacQueen, 1967),[4]
- **agg**: Hierarchical agglomerative clustering using Euclidean distance and Ward's linkage criterion (Joe H. Ward, 1963),[5]
- **agg (cos)**: The same hierarchical agglomerative clustering, but using cosine distance instead of Euclidean.

For each word pair W_1 and W_2, where W_1 is the derivational parent of W_2 and their embeddings v_1 and v_2, the clustering algorithm only gets the difference vector $d = v_2 - v_1$. The information about the word forms and their derivation type is only used in evaluation.

We evaluate the clustering quality by homogeneity (H), completeness (C) and V-measure (V) (Rosenberg and Hirschberg, 2007). These are entropy based methods, which can be compared across any number of clusters. Homogeneity is a measure of the ratio of instances of a single class pertaining to a single cluster. Completeness measures the ratio of the member of a given class that is assigned to the same cluster. V-measure is computed as the harmonic mean of homogeneity and completeness scores.

Following Gábor et al. (2017), we also report the accuracy (A) that would be achieved by the clustering if we assigned every cluster to the class that is most frequent in this cluster and then used the clustering as a classifier. The number of

[2] We count only such relations, for which both the lemmata occur at least 5 times in the Czech National Corpus.

[3] The annotators are both native speakers of Czech and they worked together in one shared document.

[4] We used standard Euclidean distance. The cosine distance does not work at all.

[5] We experiment also with other linking criteria, however, they performed much worse compared to the Ward's criterion.

POS	class	syntactic change
A→D	adjective→adverb	-ý +y, -í +ě, -ý +ě, -ý +e
A→N	designation	-ý +ec, -ý +ka
A→N	feature	-í +ost, -ý +ost
A→N	subject	-ký +tví
N→A	pertaining to	+ový, -a +ový, +ní, -a +ní, -ce +ční, +ný, +ský, -e +cký, -ka +cký
N→A	possessive	+ův, -a +in, -o +ův, -ek +kův, -a +ův
N→N	diminutization	+ek, -k +ček
N→N	instrument / scientist	-ie
N→N	man→woman	-a +ová, +ka, +ová, +vá, -ý +á, -ík +ice
N→N	man→woman / diminutization	-a +ka
N→N/A→A	super	super+
N→V	noun→verb	+ovat
V→A	ability	+elný
V→A	acting	-it +ící, -ovat +ující, -t +jící
V→A	general property	-t +vý
V→A	patient	-t +ný, -it +ený, -it +ěný, -nout +lý, -t +lý, -out +utý
V→A	purpose	-t +cí
V→N	actor	+el, -t +č
V→N	nominalization	-t +ní, -at +ání, -it +ení, -it +ění, -out +utí, -ovat +ace
V→V	imperfectivization	-at +ávat, -it +ovat
V→V	perfectivization	-at +nout, do+, na+, o+, od+, po+, pro+, pře+, při+, roz+, u+, vy+, z+, za+

Table 1: Classes of Czech derivations.

Method	cls	H	C	V	A
normalized:					
kmeans	9	67.77	56.44	**61.59**	**77.00**
agg	10	62.30	52.88	57.20	72.81
agg (cos)	8	38.90	63.06	48.12	47.48
not normalized:					
agg (cos)	8	37.93	64.97	47.90	46.38
agg	9	41.19	39.92	40.54	50.09
kmeans	7	39.92	37.38	38.61	46.92

Table 2: Comparison of different clustering methods on differences of normalized and non-normalized word-vectors trained on Czech National Corpus and clustering into 21 clusters. The results are ordered according to V-measure.

model	clust.	H	C	V	A
baseline	15	3.79	2.70	3.15	30.82
word2vec	15	75.98	57.82	**65.66**	83.06
baseline	20	5.12	3.30	4.01	31.32
word2vec	20	77.00	54.04	63.50	**84.26**
baseline	21	5.31	3.37	4.12	30.87
word2vec	21	77.50	53.17	63.07	84.12
baseline	22	5.49	3.43	4.22	30.98
word2vec	22	77.07	52.15	62.20	83.97
baseline	25	6.13	3.68	4.60	31.41
word2vec	25	80.20	53.11	63.89	87.37

Table 3: Effect of number of clusters with K-means (averaged over 10 runs).

classes (cls) shows how many classes were assigned to at least one of the clusters.

6 Results

The results on the vectors trained on Czech National Corpus and comparison of normalized and non-normalized versions are summarized in Table 2. We can see that the normalization helps both clustering methods significantly. The best method, i.e. the K-means used on the normalized word vectors is used in the next experiments.

In Table 3, we examine the effect of the number of clusters on the clustering quality. We compare our models to the baseline, in which each derivation pair is assigned to a random cluster. The table shows that regardless of the number of clusters, the clustering on the word2vec embeddings performs better than the random baseline. It shows that as we allow the K-means algorithm to form more clusters, the homogeneity increases and the completeness decreases. The V-measure is highest with the lowest number of clusters. This may be because the clusters are of uneven size. The accuracy on the word2vec model embeddings is highest around the number of clusters that corresponds to the number of classes in the data.

model	cls	H	C	V	A
word2vec	7.9	77.53	53.70	**63.45**	**84.18**
RNN dec.	6.8	73.09	52.20	60.89	83.70
RNN+a enc.	6.4	59.44	44.92	51.14	76.10
Transf. enc.	6.4	60.30	44.24	51.02	78.29
RNN+a dec.	6.8	60.94	40.25	48.48	76.40
RNN enc.	6.4	51.90	45.13	48.25	70.49
Transf. dec.	5.5	44.21	30.56	36.14	63.41
baseline	2.8	5.37	3.41	4.17	31.15
POS baseline	8	52.63	100.00	**68.97**	45.83

Table 4: Results on vectors learned by the NMT models compared to word2vec. K-means clustering with 21 clusters. The results are averaged over 10 independent runs.

Table 4 presents the results of clustering the differences of embedding vectors from NMT models. The *cls* columns shows how many different classes are assigned. Because some classes are more frequent than others, they may form the majority in multiple clusters. This is why random baseline assignes less than 3 different classes on average. We see that word2vec (trained on the Czech side of the parallel corpus) captures more information about derivations than NMT models. RNN models store more information in the embeddings if they do not utilize the attention mechanism. Even less information is stored in the embeddings by the Transformer architecture. This is probably because while in attention-less model the embedding is the only set of parameters directly associated with the given word, in the attention model the information can be split between embeddings and the attention weights. The transformer architecture with residual connections has even more parameters associated with a given word. Decoder in general stores more information about relation between words in the embeddings than encoder, presumably because it partially supplies the role of a language model.

We also evaluated clustering by POS tags (*POS baseline* in Table 4), where we created 8 clusters based on the POS tags of the parent and child words in the derivational relation. This clustering has a high V-measure, because its completeness is 100 % (the *super+* class is not present in

the NMT data and for all the other classes it holds that each member of a class has the same parent-child POS tags pair). But it has lower accuracy than all the other models (except for the random baseline), showing that the unsupervised clustering does more than just clustering by POS.

The data naturally contains classes with significant differences in size. To prevent the small classes from being underrepresented, we also evaluated the clustering on a dataset, where the same number of derivation pairs was sampled from each class. Results for the experiment with classes of the same size are listed in Table 5. The results show that the classification does not rely only on changes of part-of-speech. Both *imperfectization* and *perfectivization* classses are classified well (97 % precision, 83 % recall and 93 % precision, 66 % recall respectively), even though they are both derivation from verbs to verbs. The only classes that have both precision and recall under 50 % are those being confused with *diminutization*: *man* → *woman* shares one common derivation type with *diminutization*, and the class *super*, which contains only the prefix "super" and is therefore opposite to *diminutization*, sharing the same semantic axis.

7 Conclusion

Our results show that word-level word embeddings capture information about semantic classes of derivational relations between words, despite not having any information about the orthography or morphological makeup of the words, and therefore not knowing about the formal relation between the words.

It is possible to cluster differences between embeddings in derivational relations, and the assigned clusters correspond to the semantic classes of the relations. The word2vec embeddings generally result in a better clustering than embeddings from the NMT models, and embeddings from the decoder of a plain RNN model perform better than those from NMT models with attention. All these methods outperform a random-assignment clustering baseline and POS clustering baseline.

Acknowledgments

This work has been supported by the grant 18-02196S of the Czech Science Foundation. This study was supported by the Charles University Grant Agency (project No. 1176219). This research was partially supported by SVV project number 260 453. This work has been using language resources and tools developed, stored and distributed by the LINDAT/CLARIN project of the Ministry of Education, Youth and Sports of the Czech Republic (project LM2015071).

References

R. Harald Baayen, Richard Piepenbrock, and Leon Gulikers. 1995. The CELEX lexical database (CD-ROM).

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego*.

Ondřej Bojar, Ondřej Dušek, Tom Kocmi, Jindřich Libovický, Michal Novák, Martin Popel, Roman Sudarikov, and Dušan Variš. 2016. CzEng 1.6: Enlarged Czech-English Parallel Corpus with Processing Tools Dockered. In *Text, Speech, and Dialogue: 19th International Conference, TSD 2016*, number 9924 in Lecture Notes in Computer Science, pages 231–238, Cham / Heidelberg / New York / Dordrecht / London. Masaryk University, Springer International Publishing.

Ryan Cotterell and Hinrich Schütze. 2018. Joint semantic synthesis and morphological analysis of the derived word. *Transactions of the Association of Computational Linguistics*, 6:33–48.

Miloš Dokulil, Karel Horálek, Jiřina Hůrková, Miloslava Knappová, Jan Petr, and others. 1986. *Mluvnice češtiny (1)*, 1 edition. Academia, Prague, Czech Republic.

Kata Gábor, Haïfa Zargayouna, Isabelle Tellier, Davide Buscaldi, and Thierry Charnois. 2017. Exploring vector spaces for semantic relations. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1814–1823.

Anna Gladkova, Aleksandr Drozd, and Satoshi Matsuoka. 2016. Analogy-based detection of morphological and semantic relations with word embeddings: what works and what doesn't. In *Proceedings of the NAACL Student Research Workshop*, pages 8–15.

Pius ten Hacken. 2014. *The Oxford Handbook of Derivational Morphology*, Oxford Handbooks in Linguistics, chapter Delineating Derivation and Inflection. Oxford University Press, Oxford, United Kingdom.

Nabil Hathout and Fiammetta Namer. 2014. Démonette, a French derivational morpho-semantic network. *Linguistic Issues in Language Technology (LiLT)*, 11:125–168.

Derivation class	precision	recall
ability	70.71	68.96
acting	47.99	61.92
actor	63.03	60.00
adjective→adverb	87.72	28.00
designation	50.00	66.32
diminutization	24.09	42.40
feature	81.29	76.80
general property	63.22	69.84
imperfectivization	97.00	82.64
instrument / scientist	97.46	70.56
man→woman	98.01	62.96
man→woman / diminutization	34.88	47.52
nominalization	65.39	55.92
noun→verb	96.71	77.52
patient	47.20	51.92
perfectivization	92.85	66.48
pertaining to	36.85	52.88
possessive	65.29	78.24
purpose	52.33	31.44
subject	69.55	84.40
super	31.98	26.40

Table 5: Precision and recall for the derivation classes. We sampled 250 examples for each class from the data and clustered them with K-means on word2vec embeddings trained on the ČNK. Results presented here are averaged over 5 runs.

Jindřich Helcl, Jindřich Libovický, Tom Kocmi, Tomáš Musil, Ondřej Cífka, Dušan Variš, and Ondřej Bojar. 2018. Neural monkey: The current state and beyond. In *The 13th Conference of The Association for Machine Translation in the Americas, Vol. 1: MT Researchers' Track*, pages 168–176, Stroudsburg, PA, USA. The Association for Machine Translation in the Americas, The Association for Machine Translation in the Americas.

Milena Hnátková, Michal Kren, Pavel Procházka, and Hana Skoumalová. 2014. The syn-series corpora of written czech. In *LREC*, pages 160–164.

Jr. Joe H. Ward. 1963. Hierarchical grouping to optimize an objective function. 58(301):236–244.

Lukáš Kyjánek. 2018. Morphological resources of derivational word-formation relations. Technical Report TR-2018-61, Charles University, Faculty of Mathematics and Physics, Institute of Formal and Applied Linguistics, Prague, Czech Republic.

J. MacQueen. 1967. Some methods for classification and analysis of multivariate observations. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics*, pages 281–297, Berkeley, Calif. University of California Press.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Andrew Rosenberg and Julia Hirschberg. 2007. V-measure: A conditional entropy-based external cluster evaluation measure. In *Proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning (EMNLP-CoNLL)*.

Magda Ševčíková and Zdeněk Žabokrtský. 2014. Word-formation network for Czech. In *Proceedings of the 9th International Conference on Language Resources and Evaluation (LREC 2014)*, pages 1087–1093, Reykjavík, Iceland. European Language Resources Association.

Radu Soricut and Franz Och. 2015. Unsupervised morphology induction using word embeddings. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1627–1637.

Jana Straková, Milan Straka, and Jan Hajič. 2014. Open-Source Tools for Morphology, Lemmatization, POS Tagging and Named Entity Recognition. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 13–18, Baltimore, Maryland, USA. Association for Computational Linguistics.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008.

Britta Zeller, Sebastian Padó, and Jan Šnajder. 2014. Towards semantic validation of a derivational lexicon. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 1728–1739. Dublin City University and Association for Computational Linguistics.