The Prague Bulletin of Mathematical Linguistics
NUMBER 115 OCTOBER 2020

EDITORIAL BOARD

Published twice a year by Charles University (Prague, Czech Republic)

Editorial office and subscription inquiries:
ÚFAL MFF UK, Malostranské náměstí 25, 118 00, Prague 1, Czech Republic
E-mail: pbml@ufal.mff.cuni.cz

ISSN 0032-6585

PBML

The Prague Bulletin of Mathematical Linguistics
NUMBER 115 OCTOBER 2020

CONTENTS

Articles

Universal Derivations 1.0, A Growing Collection of Harmonised Word-Formation Resources 5
Lukáš Kyjánek, Zdeněk Žabokrtský, Magda Ševčíková, Jonáš Vidra

Inferring Highly-dense Representations for Clustering Broadcast Media Content 31
Esaú Villatoro-Tello, Shantipriya Parida, Petr Motlicek, Ondřej Bojar

Every Layer Counts: Multi-Layer Multi-Head Attention for Neural Machine Translation 51
Isaac Kojo Essel Ampomah, Sally McClean, Lin Zhiwei, Glenn Hawe

The Design of Croderiv 2.0 83
Matea Filko, Krešimir Šojat, Vanja Štefanec

Morphological Networks for Persian and Turkish: What Can Be Induced from Morpheme Segmentation? 105
Hamid Haghdoost, Ebrahim Ansari, Zdeněk Žabokrtský, Mahshid Nikravesh, Mohammad Mahmoudi

Extending Ptakopět for Machine Translation User Interaction Experiments 129
Vilém Zouhar, Michal Novák

Are Multilingual Neural Machine Translation Models Better at Capturing Linguistic Features? 143
David Mareček, Hande Celikkanat, Miikka Silfverberg, Vinit Ravishankar, Jörg Tiedemann

Derivations and Connections: Word Formation in the LiLa Knowledge Base of Linguistic Resources for Latin 163
Eleonora Litta, Marco Passarotti, Francesco Mambrini

Focalizers and Discourse Relations 187
Eva Hajičová, Jiří Mírovský, Barbora Štěpánková

Instructions for Authors 198

PBML

The Prague Bulletin of Mathematical Linguistics
NUMBER 115 OCTOBER 2020 5–30

Universal Derivations 1.0, A Growing Collection of Harmonised Word-Formation Resources

Lukáš Kyjánek, Zdeněk Žabokrtský, Magda Ševčíková, Jonáš Vidra

Charles University, Faculty of Mathematics and Physics, Institute of Formal and Applied Linguistics

Abstract

The paper deals with harmonisation of existing data resources containing word-formation features by converting them into a common file format and partially aligning their annotation schemas. We summarise (dis)similarities between the resources and describe individual steps of the harmonisation procedure, including manual annotations and application of Machine Learning techniques. The resulting "Universal Derivations 1.0" collection contains 27 harmonised resources covering 20 languages. It is publicly available in the LINDAT/CLARIAH CZ repository and can be queried via the DeriSearch tool.

1. Introduction

There are several dozens of language resources which focus specifically on derivational morphology or capture some word-formation features in addition to other types of annotation. However, the resources differ greatly in many aspects, which complicates usability of the data in multilingual projects, including potential data-oriented research in word-formation across languages.

Being inspired by the recent developments in treebanking (cf. Buchholz and Marsi, 2006, McDonald et al., 2013, Zeman et al., 2014, Nivre et al., 2016b, and others), a harmonisation procedure was proposed to unify annotation schemas of word-formation resources. The harmonised resources were released under the title *Universal Derivations* (hereafter, UDer), with eleven resources covering eleven different languages in UDer version 0.5 (Kyjánek et al., 2019; Kyjánek et al., 2019). Kyjánek (2020) elaborated on the procedure to cover resources with other data structures. The present

 Corresponding author: kyjanek@ufal.mff.cuni.cz
Cite as: Lukáš Kyjánek, Zdeněk Žabokrtský, Magda Ševčíková, Jonáš Vidra. Universal Derivations 1.0, A Growing Collection of Harmonised Word-Formation Resources. The Prague Bulletin of Mathematical Linguistics No. 115, 2020, pp. 5–30. doi: 10.14712/00326585.003.

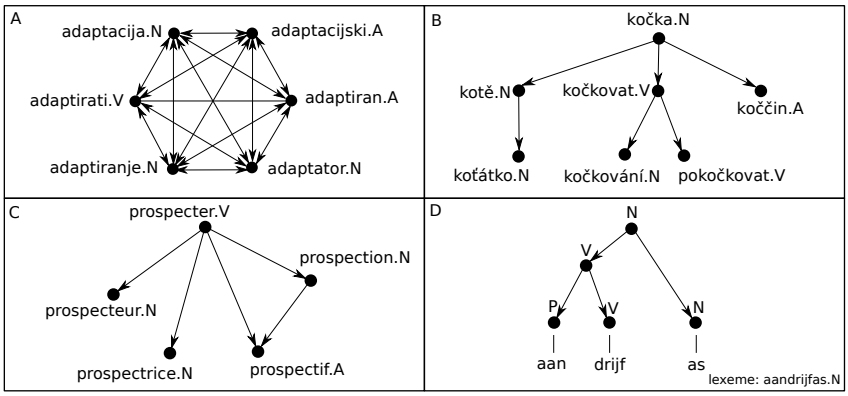


Figure 1. Data structures in available derivational resources: A. complete directed subgraph, B. rooted tree, C. weakly connected subgraph, D. derivation tree.

paper summarises the extended harmonisation procedure and introduces a new version of the UDer collection, which contains 27 harmonised resources for 20 languages (UDer 1.0, Kyjánek et al. 2020).

The paper is organised as follows: A brief overview of existing data resources, their underlying data structures, and more details on the resources selected for the harmonisation can be found in Section 2. The harmonisation is described step by step in Section 3, followed by some quantitative and qualitative features of UDer 1.0 and a description of a user query interface (Section 4).

2. Existing data resources and resources selected for harmonisation

Kyjánek (2018, 2020) listed about fifty machine-tractable resources where information related to word-formation of individual languages can be found. The resources differ in many aspects; specifically, in the data structure, in the file format, in the size in terms of both lexemes and derivational relations, and in the licenses under which the resources were released.

In what follows, the resources are compared using terms from the graph theory terminology (cf. Matoušek and Nešetřil 2009). In the first three types (see Figure 1), lexemes are represented as nodes and derivational relations as directed edges. The edges point from the base lexemes to the derived ones. In contrast, the basic building unit in the fourth type is a morpheme.

A. Some resources only group derivationally related lexemes together, i.e., lexemes that share a common root morpheme (hereafter, a derivational family). Individual derivational relations between lexemes are unspecified. Such families could be represented as complete directed subgraphs. Given that the structure mod-

els linguistic derivation, we represent such families rather by *complete directed subgraphs* (see A in Figure 1; adopted from DerivBase.hr for Croatian, Šnajder, 2014).[1]

B. If at most one base lexeme is identified for any derived lexeme, then the derivational family can be naturally represented as a *rooted tree* (B in Figure 1; from DeriNet for Czech, Vidra et al., 2019a). The tree root represents prototypically the simplest (unmotivated) lexeme, while leaf nodes contain the most complex lexemes (in terms of both morphological structure and derivational meaning) in a particular derivational family. The rooted tree data structure cannot capture compounding relations.

C. We represent derivational families as *weakly connected subgraphs* in resources that allow capturing more than one base lexeme for any derivative, e.g. compounds and double motivation (C in Figure 1; from Démonette for French, Hathout and Namer, 2014). Thus, the rooted-tree constraint does not hold in those resources.

D. Some resources focus on morphological segmentation of lexemes rather than derivational relations between lexemes. 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 (D in Figure 1; from Dutch section of CELEX2, Baayen et al., 1995); derivational relations between lexemes are then present only implicitly (based on shared sequences of morphemes).

The following criteria were applied to determine which of the existing resources will be harmonised: we wanted to cover all four data structures presented above; and we preferred resources specialised in capturing word-formation, covering languages not yet included in the collection, and published under an open license.

Bellow we briefly comment on each of the 27 resources selected for harmonisation (in alphabetical order).

[R1-en] CatVar (Habash and Dorr, 2003) is an automatically constructed Categorial Variation Database containing derivational families of English lexemes. The families were created by using the morphological segmentation obtained from several morphological segmenters and resources (including the English part of CELEX). Complete directed subgraphs are used to represent the data.

[R2-nl, R3-en, R4-de] CELEX is a large, manually created resource of comprehensive annotations for Dutch, English, and German. The three language parts were developed separately for psycholinguistic research. Word-formation features are inferred from three types of morphological segmentation provided by the resources: (a) segmentation of lexemes into bases and affixes, e.g. *collaboration* is segmented into *collaborate+ion*, (b) hierarchical segmentation of lexemes into morphemes organised into a derivation tree structure, and (c) flat segmentation of lexemes into morphemes

[1] Although keeping the quadratic number of edges in the data might seem artificial at the beginning, it is a good starting point as it allows for applying graph algorithms analogously to other types.

obtainable from the last tree level (hierarchical and flat segmentation are illustrated in example D in Figure 1).

[R5-fr] Démonette is a network containing lexemes assigned with morphological and semantic features. It was created by merging existing derivational resources for French (cf. Morphonette, Hathout, 2010; VerbAction, Tanguy and Hathout, 2002; and DériF, Namer, 2003). Démonette focuses on suffixation and is paradigm-oriented, i.e., it organises lexemes into (*partial*/*complete*) *derivational* (*sub*)*paradigms* using so-called *indirect relations*, and captures derivational series among lexemes. Derivational families are represented by weakly connected subgraphs.

[R6-cs] DeriNet is a lexical database of Czech that connects derivationally related lexemes. The data format used since version 2.0 (Vidra et al., 2019b) allows to represent compounding and other features, such as morphological categories, morphological segmentation, semantic labels, etc. Each derivational family is represented as a rooted tree.

[R7-es] DeriNet.ES (Faryad, 2019) is a DeriNet-like lexical database for Spanish. Its derivational relations were created by using substitution rules covering Spanish affixation. Resulting derivational families are organised into rooted trees.

[R8-fa] DeriNet.FA (Haghdoost et al., 2019) is a lexical database capturing derivations in Persian. It was created on top of the manually compiled Persian Morphologically Segmented Lexicon (Ansari et al., 2019). Derivationally related lexemes were identified and organised into DeriNet-like rooted trees by using automatic methods.

[R9-it] DerIvaTario (Talamo et al., 2016) is a database of manually morphologically segmented Italian lexemes. Each lexeme is assigned a unique ID, which interconnects lexemes across several existing resources to provide various pieces of information such as morphological categories and phonetic transcriptions. The data is processed as derivation tree structures.

[R10-de] DErivBase (Zeller et al., 2013) is a large-coverage lexicon for German, in which derivational relations were identified by more than 190 derivational rules, i.e., string substitutions, extracted from German reference grammar books. The resulting derivational families were automatically split into semantically consistent clusters by Zeller et al. (2014), forming weakly connected subgraphs.

[R11-hr] DerivBase.Hr is a database containing Croatian derivational families. Inspired by German DErivBase and DErivCELEX (Shafaei et al., 2017), DerivBase.Hr was created by using a set of derivational rules. Since the resource lists derivational families without specifying individual derivational relations, we represent the data as complete directed subgraphs.

[R12-ru] DerivBase.Ru (Vodolazsky, 2020) is a data resource of Russian derivationally related lexemes. While its lexemes came from Russian Wikipedia and Wiktionary, the relations were identified by a set of derivational rules extracted from Russian grammar books. Derivational families are represented as weakly connected subgraphs.

[R13-et] EstWordNet (Kerner et al., 2010) is an Estonian WordNet-like lexical database, into which derivational relations were added by Kahusk et al. (2010). The resulting families are represented as weakly connected subgraphs.

[R14-ca, R15-cs, R16-gd, R17-pl, R18-pt, R19-ru, R20-sh, R21-sv, R22-tr] Etymological WordNet (Gerard, 2014) is a lexical resource containing data extracted from the English section of Wiktionary. The Etymological WordNet aims, differently from other wordnets, at identifying lexemes linked by etymology. Besides etymological features, the Etymological WordNet also captures derivational relations between lexemes for almost 180 languages; however, only a few relations are captured for many languages. The data is mostly represented as weakly connected subgraphs.

[R23-fi] FinnWordNet (Lindén and Carlson, 2010) is a WordNet-like database created by translating English WordNet into Finnish. Derivational relations were added by Lindén et al. (2012). Derivational families are represented as weakly connected subgraphs.

[R24-pt] NomLex-PT (De Paiva et al., 2014) is a lexicon of Brazilian Portuguese verbs and deverbative nouns, which were extracted from already existing resources. Resulting derivational families are represented as weakly connected subgraphs.

[R25-en] The Morpho-Semantic Database (Fellbaum et al., 2007) is a stand-off database linking morphologically related nouns and verbs from English (Princeton) WordNet version 3.0 (Miller, 1998). Derivational relations were identified automatically and assigned 14 semantic labels. The data is represented as weakly connected subgraphs.

[R26-pl] The Polish Word-Formation Network (Lango et al., 2018) is a DeriNet-like lexical network for Polish. It was created by using pattern-mining techniques and a machine-learned ranking model. The Polish WFN was also enlarged with derivational relations extracted from the Polish WordNet (Maziarz et al., 2016). Derivational families are represented as rooted trees.

[R27-la] Word Formation Latin (Litta et al., 2016) is a manually-annotated resource specialised in capturing word-formation of Latin. Its lexeme set is based on the Oxford Latin Dictionary (Glare, 1968). In the Word Formation Latin database, the majority of derivational families is represented as rooted trees, but weakly connected subgraphs are used to capture compounds.

3. Harmonisation procedure

Once the resources were selected and their data structures identified, the harmonisation procedure started, focusing on both the data (i.e., lexeme set, derivational relations, and annotated features) and the annotation schemas (i.e., data structure, file format, feature-value pairs) of the resources.

Concerning the data, we have decided to make as few changes as possible. Thus, (i) the original sets of (derivationally related) lexemes are neither enlarged nor reduced; (ii) all derivational relations from the input resources are still preserved in

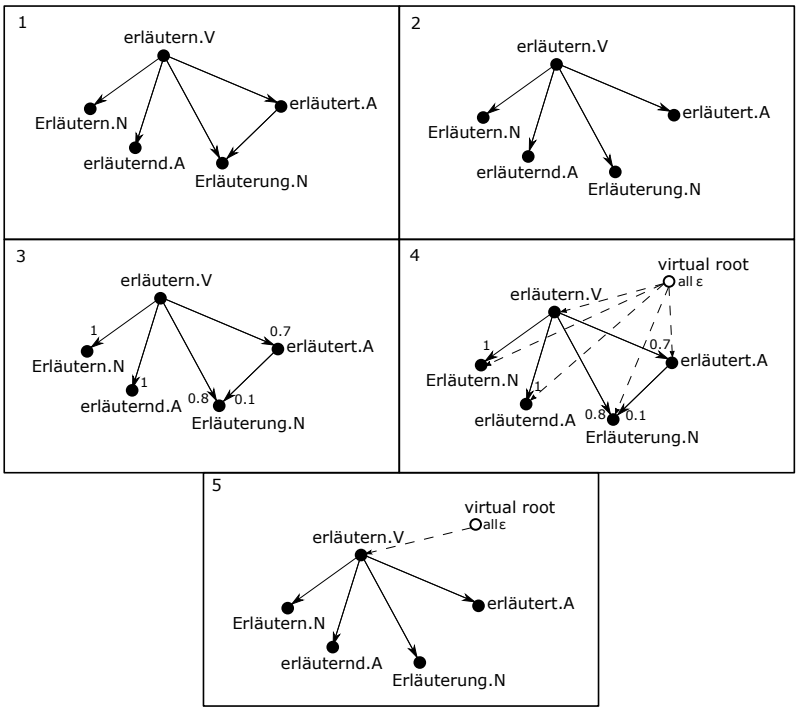


Figure 2. Five steps of the harmonisation procedure (illustrated on DErivBase data).

the resulting data, although they are restructured to fit the selected target annotation schema; and (iii) no new features or pieces of annotations are added to the data.

As for the annotation schema, we have selected the rooted-tree data structure and the file format used in DeriNet 2.0 (Vidra et al., 2019b) as the target data representation for all the resulting harmonised resources. In DeriNet 2.0, each derivational family is represented as a rooted tree (e.g. example B in Figure 1), which is internally organised according to the morphemic complexity of the lexemes involved, from the morphematically simplest lexeme in the root of the tree to the most complex ones in the leaves. Thus, it concurs with the linguistic account of derivation as a process of adding an affix to a base in order to create a new lexeme (Dokulil, 1962; Iacobini, 2000; Lieber and Štekauer, 2014). This simple but, at the same time, highly constrained rooted-tree data structure makes it possible to store massive amounts of language data in a unified way, but it is not sufficient for capturing compounding and other more intricate phenomena, such as double motivation. These issues have been solved by introducing secondary edges allowing to specify any number of base lexemes and

derivatives in the target data structure.[2] We believe that such a representation is a reasonable compromise between expressiveness and uniformity. In addition, choosing the tree approach is hard to resist from the practical perspective: it simplifies many technical aspects (compared to less constrained graphs), such as data traversing and visualisation.

The target file format, in which the target data structure is stored, is a textual lexeme-based format consisting of ten tab-separated columns (Vidra et al., 2019b, pp. 86-88), inspired by the CoNLL-U format (Nivre et al., 2016a) used in Universal Dependencies treebanks. It has been designed to be as universal as possible to allow preserving key-value pairs specifying most of the annotated features relevant for studying word-formation, such as part-of-speech or any morphological categories of lexemes. The list of features can be extended as needed for any language, and lexeme features which cannot be easily expressed by a single value can also be stored in JSON format in the last column of the file.

The harmonisation procedure is illustrated in steps 1 through 5 in Figure 2 and further described in Subsections 3.1 to 3.5, respectively. Each of the steps is exemplified on two German resources (G-CELEX and DErivBase) in order to provide a better insight.

3.1. Importing data from the existing resources

At the beginning of the procedure, we import as much information as possible from the original, resource-specific file formats of the input resources. For instance, the DErivBase file format lists all lexeme pairs within the same derivational family and the shortest path between any two lexemes in the family (the top of Figure 3). The paths consist of derivational relations to which so-called *derivational rules* are assigned, e.g. `dVN09*>`.[3] In the case of G-CELEX, its file format stores individual lexemes with three types of morphological segmentation without specifying derivational relations between lexemes (the bottom of Figure 3).

First, we import lexemes. In most of the resources, a lexeme is represented as its lemma accompanied with its part-of-speech tag. In addition to lemma and part-of-speech tag, gender is used for representing nouns in DErivBase, while only a unique numeric ID is used in G-CELEX. We import only derivationally related lexemes from Estonian, Finnish, and Etymological WordNets, disregarding synonymy relations and the hyponymic/hyperonymic architecture completely.

After obtaining the lexeme sets, other pieces of annotations are imported, e.g. derivational and compounding relations between lexemes, morphological categories and

[2] This aspect resembles the case of Universal Dependencies, where it was also clear from the very beginning that trees are insufficient for capturing all syntactic relations (e.g. with more complex coordination expressions). The recent UD solution is that for each sentence, there is a core tree-shaped structure, possibly accompanied with a set of secondary (non-tree) edges.

[3] The asterisk (*) indicates that the rule is applied inversely.

```
Erläutern_Nn erläutern_V 1 Erläutern_Nn dVN09*> erläutern_V
Erläutern_Nn erläuternd_A 2 Erläutern_Nn dVN09*> erläutern_V dVA12> erläuternd_A
Erläutern_Nn erläutert_A 2 Erläutern_Nn dVN09*> erläutern_V dVA13> erläutert_A
Erläutern_Nn Erläuterung_Nf 2 Erläutern_Nn dVN09*> erläutern_V dVN07> Erläuterung_Nf
erläutern_V erläuternd_A 1 erläutern_V dVA12> erläuternd_A
erläutern_V erläutert_A 1 erläutern_V dVA13> erläutert_A
erläutern_V Erläuterung_Nf 1 erläutern_V dVN07> Erläuterung_Nf
erläuternd_A erläutert_A 2 erläuternd_A dVA12*> erläutern_V dVA13> erläutert_A
erläuternd_A Erläuterung_Nf 2 erläuternd_A dVA12*> erläutern_V dVN07> Erläuterung_Nf
erläutert_A Erläuterung_Nf 1 erläutert_A dNA25> Erläuterung_Nf
```

```
1\Tourenschi\...\Tour+en+Schi\NxN\...\((Tour)[N],(en)[N|N.N],(Schi)[N])[N]\...
2\Tourenwagen\...\Tour+en+Wagen\NxN\...\((Tour)[N],(en)[N|N.N],(Wagen)[N])[N]\...
3\tourenweise\...\Tour+en+weise\Nxx\...\((Tour)[N],(en)[B|N.x],(weise)[B|Nx.])[B]\...
4\Tourenzahl\...\Tour+en+zaehl\NxV\...\((Tour)[N],(en)[N|N.V],(zaehl)[V])[N]\...
5\Tourenzaehler\...\Tour+en+zaehl+er\NxVx\...\((Tour)[N],(en)[N|N.Vx],(zaehl)[V],(er)[N|NxV.])[N]\...
6\Tourismus\...\Tour+ismus\Nx\...\((Tour)[N],(ismus)[N|N.])[N]\...
7\Tourist\...\Tour+ist\Nx\...\((Tour)[N],(ist)[N|N.])[N]\...
8\Touristik\...\Tour+istik\Nx\...\((Tour)[N],(istik)[N|N.])[N]\...
9\touristisch\...\Tour+istisch\Nx\...\((Tour)[N],(istisch)[A|N.])[A]\...
```

Figure 3. Original file formats of DErivBase (top) and G-CELEX (bottom).

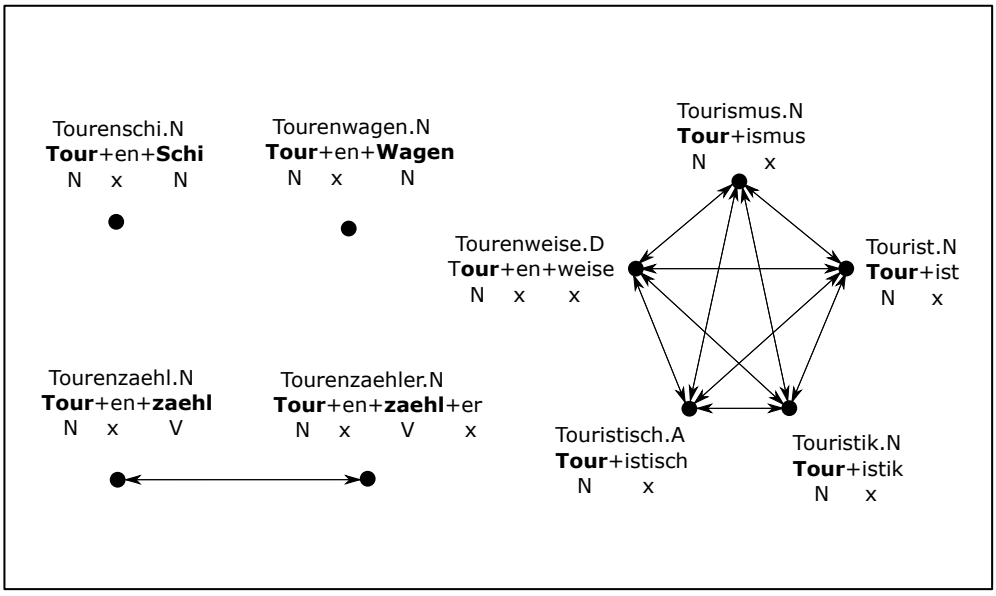


Figure 4. Constructing derivational subgraphs from morphological segmentation.

Input resource	Imported features from the input data resources									
	DER	COM	POS	MCG	SEG	SEM	HSG	PAR	TID	DRL
[R1-en] CatVar	✓	–	✓	–	–	–	–	–	–	–
[R2-nl] D-CELEX	–	✓	✓	–	✓	–	✓	–	✓	–
[R3-en] E-CELEX	–	✓	✓	–	✓	–	✓	–	✓	–
[R4-de] G-CELEX	–	✓	✓	–	✓	–	✓	–	✓	–
[R5-fr] Démonette	✓	–	✓	✓	✓	✓	–	✓	–	–
[R6-cs] DeriNet	✓	✓	✓	✓	✓	✓	–	–	✓	–
[R7-es] DeriNet.ES	✓	–	–	–	–	–	–	–	–	–
[R8-fa] DeriNet.FA	✓	–	–	–	–	–	–	–	–	–
[R9-it] DerIvaTario	–	–	✓	–	✓	–	–	–	✓	–
[R10-de] DErivBase	✓	–	✓	✓	–	–	–	–	–	✓
[R11-hr] DerivBase.Hr	✓	–	✓	–	–	–	–	–	–	–
[R12-ru] DerivBase.Ru	✓	–	✓	–	–	–	–	–	–	✓
[R13-et] EstWordNet	✓	–	✓	–	–	–	–	–	–	–
[R14-ca] EtymWordNet-cat	✓	–	–	–	–	–	–	–	–	–
[R15-cs] EtymWordNet-ces	✓	–	–	–	–	–	–	–	–	–
[R16-gd] EtymWordNet-gla	✓	–	–	–	–	–	–	–	–	–
[R17-pl] EtymWordNet-pol	✓	–	–	–	–	–	–	–	–	–
[R18-pt] EtymWordNet-por	✓	–	–	–	–	–	–	–	–	–
[R19-ru] EtymWordNet-rus	✓	–	–	–	–	–	–	–	–	–
[R20-sh] EtymWordNet-hbs	✓	–	–	–	–	–	–	–	–	–
[R21-sv] EtymWordNet-swe	✓	–	–	–	–	–	–	–	–	–
[R22-tr] EtymWordNet-tur	✓	–	–	–	–	–	–	–	–	–
[R23-fi] FinnWordNet	✓	–	✓	–	–	–	–	–	–	–
[R24-pt] NomLex-PT	✓	–	✓	–	–	–	–	–	–	–
[R25-en] The M-S Database	✓	–	✓	–	–	✓	–	–	–	–
[R26-pl] The Polish WFN	✓	–	–	–	–	–	–	–	–	–
[R27-la] Word Formation Latin	✓	✓	✓	✓	✓	–	–	–	✓	–

Table 1. Features imported from the individual data resources: derivational relations (DER), compounding relations (COM), part-of-speech tags (POS), morphological categories (MCG), morphological segmentation (SEG), semantic labels (SEM), hierarchical segmentation (HSG), subparadigmatic relations (PAR), unique technical IDs of lexemes (TID), derivational rules (DRL).

morphological segmentation, semantic labels, etc. We also extract custom features, such as derivational rules from DErivBase and hierarchical morphological segmentation from G-CELEX; see Table 1.

Based on the imported relations and features, we identify the original data structure type for each family; see step 1 in Figure 2. The original data for the particular family is presented at the top of Figure 3. In derivation tree structures, where the relations between lexemes are not captured, e.g. in G-CELEX, we generate relations on the basis of the shared (root) morphemes and longer subsequences of morphemes in the lexemes.[4] For instance, derivational relations are generated for *Tour+ist* '*tourist*' on the basis of `Nx` representing suffixation of the base *Tour* (cf. line 5 and 7 in the bottom of Figure 3). We generate compounding relations too, e.g. for *Tour+en+Wagen* '*tour-*

[4]Homonymy of morphemes is a difficult problem to solve here.

ing car' segmented as `NxN` (cf. line 2 in the bottom of Figure 3). However, we do not apply further harmonisation steps to them. In the case of generated derivational relations, we obtain derivational families represented as complete or weakly connected subgraphs (see Figure 4), in which the target rooted-tree data structures have to be identified, if the particular family is not already tree-shaped.

For DeriNet, DeriNet.ES, DeriNet.FA, and the Polish WFN, which contain rooted trees as their original data structure, the following steps 2, 3, and 4 are unnecessary, and the resources are included into the resulting collection by skipping to the last step of the whole procedure (Section 3.5).

3.2. Annotating derivational families

As the next step in harmonisation of non-tree resources, we have identified rooted trees of non-tree-shaped families. Most resources contain only a handful of such families (see Table 2), making it possible to identify the rooted tree manually in all of them. However, CatVar, D-CELEX, E-CELEX, G-CELEX, DErivBase, DerivBase.Hr, DerivBase.Ru and FinnWordNet contain too many non-trees to be handled by hand. In such resources, we have manually annotated a uniformly random sample of 400 to 600 derivational families, which served as training and testing data for development of supervised Machine Learning models.

In all non-tree-shaped derivational families, the annotators' task was to choose derivational relations which form a tree-shaped structure (see step 2 in Figure 2). During the annotations, annotators[5] were not allowed to add any new lexemes and relations. The phenomena on which the annotators decided are exemplified in Figure 5. In tree A (from DErivBase), both the adjective *stehend* '*standing*' and the verb *nachstehen* '*lag behind*' were considered as base lexemes for the adjective *nachstehend* '*lagging behind*' because they share a long common substring, but the verb was chosen as the linguistically adequate solution as *nachstehend* is a present participle form of this verb and is not assumed to be a prefixation of another participle (*stehend*). In contrast, in B in Figure 5 two representations seem to be equally acceptable: either two parallel subtrees are constructed (one for affirmatives *gelenkig* '*flexible*' and *Gelenkigkeit* '*flexibility*', the second one for negatives *ungelenkig* '*inflexible*' and *Ungelenkigkeit* '*inflexibility*'), or negated lexemes are directly linked with their affirmative forms, i.e., *gelenkig* → *ungelenkig* and *Gelenkigkeit* → *Ungelenkigkeit*. We chose the latter solution because it keeps the trees more compact and is insensitive to missing lexemes as compared to the former option, and applied it across the harmonised resources.

[5]We annotated most of the resources ourselves using several monolingual and multilingual dictionaries. In the case of DerivBase.Ru, the annotator was a Russian native speaker and a linguist at the same time.

Input resource	Tree-shaped families	Tree-shaped relations	Non-tree-shaped families	Non-tree-shaped relations	Manually annotated families	Manually annotated relations
[R1-en] CatVar	0	0	13,367	155,064	600	7,618
[R2-nl] D-CELEX	0	0	5,449	1,733,364	419	6,596
[R3-en] E-CELEX	0	0	6,725	109,002	411	6,654
[R4-de] G-CELEX	0	0	5,615	145,936	449	5,720
[R5-fr] Démonette	7,050	12,849	286	1,303	286	1,303
[R9-it] DerIvaTario	0	0	1,992	28,088	440	5,454
[R10-de] DErivBase	15,831	21,795	3,962	33,215	431	5,226
[R11-hr] DerivBase.Hr	0	0	14,818	3,056,962	610	7,548
[R12-ru] DerivBase.Ru	7,653	10,076	10,293	279,817	455	10,754
[R13-et] EstWordNet	428	470	28	65	28	65
[R14-ca] EtymWordNet-cat	2,879	4,422	40	191	40	191
[R15-cs] EtymWordNet-ces	2,284	4,788	70	543	70	543
[R16-gd] EtymWordNet-gla	2,412	4,688	57	403	57	403
[R17-pl] EtymWordNet-pol	2,822	24,106	59	879	59	879
[R18-pt] EtymWordNet-por	1,166	1,586	15	41	15	41
[R19-ru] EtymWordNet-rus	715	2,926	36	474	36	474
[R20-sh] EtymWordNet-hbs	1,694	6,111	20	238	20	238
[R21-sv] EtymWordNet-swe	2,865	4,075	20	376	20	376
[R22-tr] EtymWordNet-tur	1,837	5,188	84	769	84	769
[R23-fi] FinnWordNet	2	2	6,345	29,781	377	2,432
[R24-pt] NomLex-PT	2,751	4,124	34	111	34	111
[R25-en] The M-S Database	5,690	7,580	128	420	128	420
[R27-la] Word Formation Latin	5,230	21,946	43	741	43	741

Table 2. Number of tree-shaped and non-tree-shaped families in the input resources and the size of manually annotated samples. Structures consisting of a single lexeme (so-called singletons), and relations explicitly labelled as compounding are not considered.

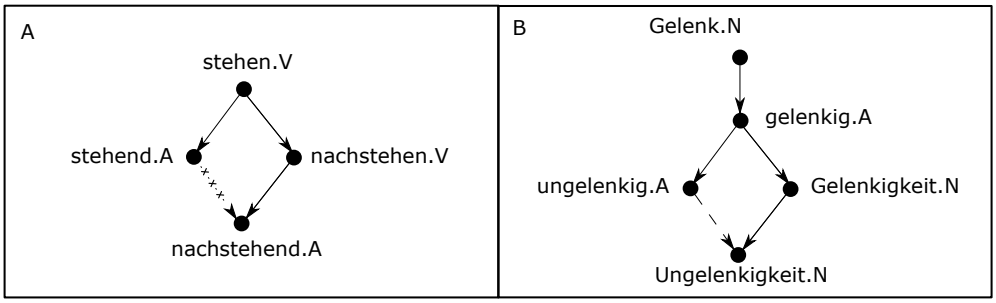


*Figure 5. Manual annotation (DErivBase): A. The prefixed adjective (*nachstehend*) captured as derived from the prefixed verb (vs. the rejected relation represented by the dotted line with tiny crosses). B. The noun with a negative prefix (*Ungelenkigkeit*) can be seen as a deadjectival derivative (from* ungelenkig*; cf. the dashed line) or a denominal derivative (*Gelenkigkeit*); the latter representation is preferred in UDer.*

3.3. Scoring derivational relations

The above-mentioned manual annotation was aimed at selecting a tree-shaped subgraph out of the original resource. Given the annotated data, we want to automatise this task for all families using Machine Learning.

From the Machine Learning perspective, the task can be formalised in various ways. We choose an approach consisting of two phases:

1. We train a scoring model that assigns a numerical score to each edge; the higher the score, the higher the chance that a given edge belongs to the rooted tree.
2. We choose the rooted tree with the maximum sum of edge scores.

We tackle only the first phase using Machine Learning, as described in the following paragraphs. Once the edge scores are given, the globally optimal rooted tree can be found deterministically in the second phase, as described in more detail in Section 3.4.

Manually annotated data does not provide us with any numerical values to train a scorer directly. What we have in each annotated family are relations that were manually marked to be included into the tree (positive examples), while all the other relations from the original data resources are considered as rejected (negative examples). Using this view, we can reformulate the scoring task as a classification task: we train a binary classifier that predicts each relation to be accepted or rejected. Then we use the classifier's confidence about the positive class as the score.

The classification data was prepared as follows: we split the annotated data into the training (65 %), validation (15 %), and hold-out (20 %) sections, and provided all positive and negative instances with the following one-hot encoded features: (a) part-of-speech categories, (b) morphological categories, such as gender, aspect, etc., if present in the original resource, (c) initial and final character n-grams of both the base lexeme and derivative, (d) custom features included in particular resources, e.g. derivational rules in DerivBase.Ru; and of the following numeric features: (e) Levenshtein distance (Levenshtein, 1966), (f) Jaro-Winkler distance (Jaro, 1989; Winkler, 1990), (g) Jaccard distance (Jaccard, 1912), and (h) length of the longest common substring.

Table 3 summarises performance of the following classification methods: Naive Bayes, Logistic Regression, Decision Tree, Random Forest, Perceptron, and K-Nearest Neighbour. Clearly, it was necessary to train a separate classification model for each data resource. If hyper-parameter settings were needed, the values were set using grid-search on the training and validation sections. The standard classification methods are compared with a simple probabilistic baseline whose score is a maximum-likelihood conditional probability estimation conditioned only by the pair of POS values of related lexemes.

Finally, two things should be emphasised. First, the achieved performances are not directly comparable across different data resources, as the complexity of the particular classification tasks might be highly different. Second, the classification performance

Resource	ML method	ε	Scoring relations VALIDATION	Scoring relations HOLDOUT	Identifying trees VALIDATION	Identifying trees HOLDOUT
[R1-en] CatVar	Decision Tree	0.5	44.6 / 82.4	44.9 / 80.7	51.6 / 83.1	53.3 / **81.0**
[R2-nl] D-CELEX	Decision Tree	0.3	47.2 / 81.1	47.7 / 77.1	54.2 / 81.1	53.0 / **79.5**
[R3-en] E-CELEX	Decision Tree	0.5	47.1 / 74.0	47.1 / 74.0	59.7 / 74.9	59.4 / **73.8**
[R4-de] G-CELEX	Decision Tree	0.5	45.8 / 75.6	46.1 / 76.8	57.5 / 79.5	57.5 / **77.4**
[R9-it] DerIvaTario	Decision Tree	0.6	47.7 / 77.5	47.5 / 76.0	48.7 / 78.1	50.0 / **75.1**
[R10-de] DErivBase	Logistic Regression	0.1	24.9 / 88.6	25.4 / 85.8	75.1 / 93.4	78.9 / **92.1**
[R11-hr] DerivBase.Hr	Decision Tree	0.2	45.2 / 77.2	45.4 / 80.7	56.4 / 81.1	58.3 / **81.0**
[R12-ru] DerivBase.Ru	Logistic Regression	0.0	35.1 / 83.0	34.1 / 83.1	49.3 / 84.4	45.0 / **85.5**
[R23-fi] FinnWordNet	Random Forest	0.3	38.2 / 74.0	37.8 / 70.1	62.0 / 80.2	62.9 / **76.9**

Table 3. Evaluation of F-scores calculated for harmonisation procedure that uses simple baseline vs. Machine Learning model (in form: `simple_baseline / ml_model`).

should be considered only a proxy measure and cannot be assumed to correlate perfectly with the quality of induced rooted trees.

3.4. Constructing rooted trees

We construct the resulting score-optimal rooted tree on top of a derivational family with scored relations using the Maximum Spanning Tree (MST) algorithm (Chu and Liu, 1965; Edmonds, 1967). If any rooted tree exists in the input graph, then this algorithm is guaranteed to find the rooted tree with maximum sum of scores, see step 4 in Figure 2.

However, rooted trees are not guaranteed to exist in derivation families imported from the input resources. In order to make sure that the MST algorithm will not fail, we add a temporary virtual root into each family and connect it with all lexemes in the family (see Figure 6.). The score of such newly added relations is equal to ε; the optimal value of ε is grid-searched using the validation sections for each resource separately.

Besides guaranteeing that the MST algorithm will not fail, adding a virtual root with ε-scored relations makes it possible to effectively split a family into two or more disconnected components, as the virtual root is deleted at the end. However, all roots of the divided family are still interlinked in the last JSON-encoded column in the target file format of the resulting harmonised data, e.g. roots *betreffen* '*to affect*' and *Betreff* '*subject*' in Figure 6.

The performances of identification of rooted trees applied to data predicted by both the simple baseline models and the Machine Learning models are presented in Table 3; the bold numbers show the final performances of the whole harmonisation procedure.

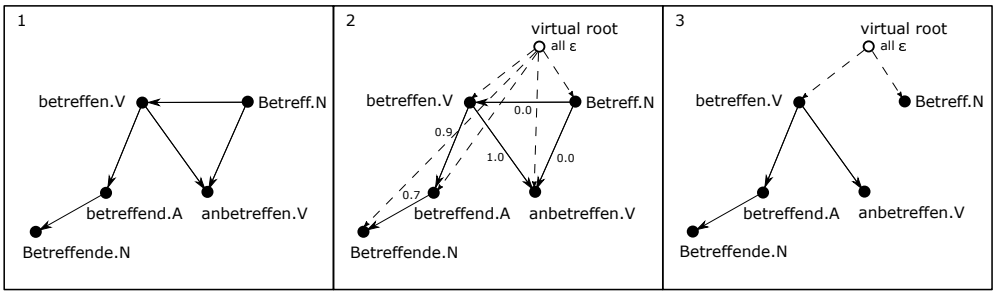


Figure 6. Identification of rooted trees by maximising the sum of scores (DErivBase).

3.5. Converting data into the DeriNet 2.0 format

Finally, we convert the identified rooted trees (except for the virtual root and its relations; cf. step 5 in Figure 2) into the target DeriNet 2.0 file format using the application interface developed for DeriNet 2.0.[6]

We preserve all relations from the original data resources, including compounding relations and relations not chosen by the MST algorithm, just that these additional relations are stored in a less prominent place in the target file format.[7]

We also convert all (custom) features assigned to the lexemes (e.g. part-of-speech categories, morphological categories, morphological segmentation, etc.) and relations, such as semantic labels or derivational rules. Part-of-speech values and morphological categories are also harmonised using the Universal Features annotation schema (Nivre et al., 2016a); however, values of semantic labels are kept in their original forms because they differ significantly across the resources.

We convert the lemma set of each resource and all features assigned to the lexemes first. A unique identifier for each lexeme is created to prevent technical problems caused by the same string form. For example, DErivBase uses a combination of the lemma, its part-of-speech tag, and its gender (for nouns), but G-CELEX combines the lemma and its part-of-speech tag with the original numeric ID to disambiguate lexemes.

After that, we convert tree-shaped derivational relations and add their annotations, for instance, derivational rules from DErivBase. They are stored under the key `Rule=`*x*, where *x* is the original rule identifier. In G-CELEX, a complete (hierarchical) morphological segmentation as well as compounding relations are also included.

[6] `https://github.com/vidraj/derinet/tree/master/tools/data-api/derinet2`

[7] However, we do not preserve non-tree relations in the harmonised versions of CatVar and DerivBase.Hr. It would be too redundant, as we represent their derivational families as complete directed subgraphs initially.

```
1.0 erläutern#VERB  erläutern VERB  _ _ _ _ _ {}
1.1 Erläutern#NOUN#Neut Erläutern NOUN  Gender=Neut _ 1.0 Rule=dVN09&Type=Deriv _ {}
1.2 Erläuterung#NOUN#Fem  Erläuterung NOUN  Gender=Fem  _ 1.0 Rule=dVN07&Type=Deriv _ {}
1.3 erläuternd#ADJ  erläuternd  ADJ _ _ 1.0 Rule=dVA12&Type=Deriv _ {}
1.4 erläutert#ADJ erläutert ADJ _ _ 1.0 Rule=dVA13&Type=Deriv _ {"other_parents": "1.2&Rule=dNA25"}
```

Figure 7. A derivational family from DErivBase harmonised in the target file format.

Finally, we add some other information, such as the original non-tree derivational relations excluded during the harmonisation and links between tree roots if an original family was divided after the identification of rooted trees. [8] These annotations are stored in the last JSON-encoded column in the target file format.

Figure 7 presents a derivational family from DErivBase harmonised to the final file format. The meaning of individual columns is as follows: (i) internal ID consisting of the word-formation family number and the lexeme number separated by a dot, (ii) unique identifier for each lexeme, (iii) lemma, (iv) part-of-speech tag, (v) morphological features, (vi) surface morphological segmentation, (vii) ID of the base lexeme, (viii) annotations of the relation referenced to by the internal ID, (ix) column reserved for other potential relations, (x) JSON-encoded data.

4. Universal Derivations collection

The resulting collection, Universal Derivations version 1.0 (UDer 1.0), contains 27 resources covering 20 languages; see Table 4 summarising basic characteristics of the collection and Figure 8 with examples of harmonised trees. If a particular language is covered by more resources in the collection, the same lexeme was chosen, cf. the English verb *to abandon* in Catvar, E-CELEX, and the Morpho-Semantic Database, or the Russian noun ***весна*** '*spring*' with different derivatives and different relations in DerivBase.Ru and EtymWordNet-rus. The tree of the Polish verb *chcieć* '*to want*' in the Polish WFN differs from the EtymWordNet-pol tree in that it also includes (inflected) word forms.

4.1. Selected quantitative and qualitative properties

Selected quantitative and qualitative details on the UDer 1.0 collection are documented in Table 4 and commented in the following subsections.

Lex(emes). The lexeme sets are adopted from the original data resources, except for EstWordNet, FinnWordNet, and Etymological WordNets from which only derivationally related lexemes are taken. Multi-word lexemes (used mostly for phrasal verbs

[8] They are not preserved for the harmonised versions of CatVar and DerivBase.Hr, because we represented their families as complete (directed) subgraphs at the beginning of the procedure.

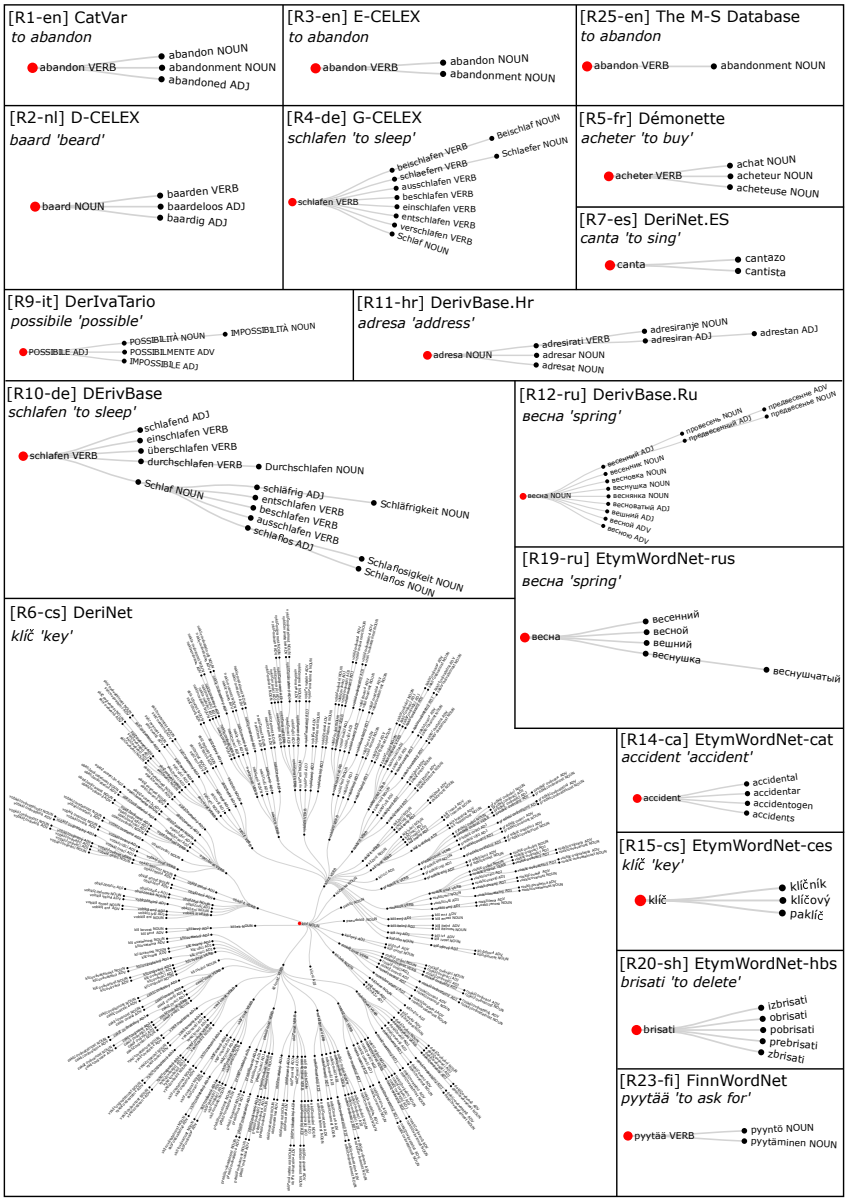


Figure 8. Harmonised rooted trees in UDer 1.0 (part 1).

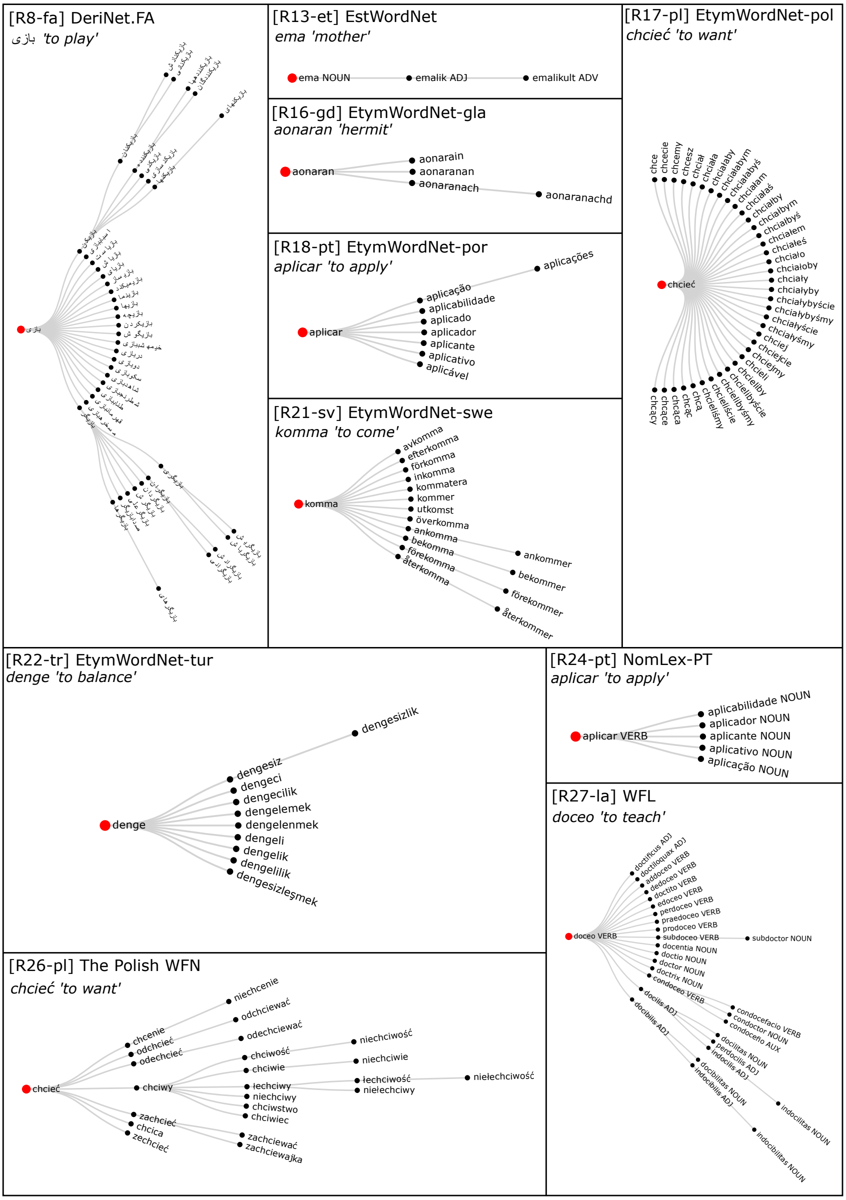


Figure 8. Harmonised rooted trees in UDer 1.0 (part 2).

Lang & Resource	Lex	Rel	Fam	Singl	Size	Depth	Out-deg	POS distrib.	License
[R1-en] CatVar	82,675	24,873	57,802	45,954	1.4/18	0.3/7	0.3/10	60/24/11/5/0	OSL-1.1
[R2-nl] D-CELEX	125,611	13,435	112,176	107,112	1.1/301	0.1/11	0.1/73	63/8/8/1/21	–
[R3-en] E-CELEX	53,103	9,826	43,277	37,951	1.2/51	0.2/8	0.2/33	47/15/13/7/18	–
[R4-de] G-CELEX	53,282	13,553	39,729	34,156	1.3/39	0.2/11	0.3/35	52/17/17/2/12	–
[R5-fr] Démonette	21,290	13,808	7,482	69	2.8/12	1.1/4	1.8/8	63/2/34/0/0	C +NC 3.0
[R6-cs] DeriNet	1,027,665	809,282	218,383	96,208	4.7/1638	0.8/10	1.1/40	44/35/5/16/0	C +NC 3.0
[R7-es] DeriNet.ES	151,173	36,935	114,238	98,325	1.3/35	0.2/5	0.3/14	0/0/0/0/0	C +NC 3.0
[R8-fa] DeriNet.FA	43,357	35,745	7,612	0	5.7/180	1.5/6	3.3/114	0/0/0/0/0	C +NC 4.0
[R9-it] DerIvaTario	8,267	1,787	6,480	5,255	1.3/13	0.2/5	0.2/6	51/26/14/9/0	C 4.0
[R10-de] DErivBase	280,775	43,368	237,407	216,982	1.2/46	0.1/5	0.1/13	86/10/5/0/0	C 3.0
[R11-hr] DerivBase.Hr	99,606	35,289	64,317	50,100	1.5/945	0.3/21	0.4/863	59/30/12/0/0	C 3.0
[R12-ru] DerivBase.Ru	270,473	133,759	136,714	116,037	2.0/1142	0.3/13	0.4/36	62/18/17/3/0	Apache 2.0
[R13-et] EstWordNet	988	507	481	22	2.1/3	1.0/2	1.0/3	16/29/8/47/0	C 3.0
[R14-ca] EtymWordNet-cat	7,496	4,568	2,928	8	2.6/13	1.1/4	1.5/13	0/0/0/0/0	C 3.0
[R15-cs] EtymWordNet-ces	7,633	5,237	2,396	14	3.2/48	1.1/4	2.0/42	0/0/0/0/0	C 3.0
[R16-gd] EtymWordNet-gla	7,524	5,013	2,511	15	3.0/15	1.1/3	1.8/13	0/0/0/0/0	C 3.0
[R17-pl] EtymWordNet-pol	27,797	24,876	2,921	19	9.5/75	1.1/3	8.3/66	0/0/0/0/0	C 3.0
[R18-pt] EtymWordNet-por	2,797	1,610	1,187	9	2.4/57	1.0/3	1.3/57	0/0/0/0/0	C 3.0
[R19-ru] EtymWordNet-rus	4,005	3,227	778	15	5.1/44	1.0/3	4.0/44	0/0/0/0/0	C 3.0
[R20-sh] EtymWordNet-hbs	8,033	6,303	1,730	6	4.6/108	1.0/3	3.6/107	0/0/0/0/0	C 3.0
[R21-sv] EtymWordNet-swe	7,333	4,423	2,910	3	2.5/116	1.0/3	1.5/116	0/0/0/0/0	C 3.0
[R22-tr] EtymWordNet-tur	7,774	5,837	1,937	11	4.0/42	1.1/4	2.8/22	0/0/0/0/0	C 3.0
[R23-fi] FinnWordNet	20,035	11,922	8,113	1,461	2.5/20	1.0/5	1.3/14	55/29/15/0/0	C 4.0
[R24-pt] Nomlex-PT	7,020	4,201	2,819	17	2.5/7	1.0/1	1.5/7	60/0/40/0/0	C 4.0
[R25-en] The M-S Database	13,813	7,855	5,958	65	2.3/6	1.0/1	1.3/6	57/0/43/0/0	C +NC 3.0
[R26-pl] The Polish WFN	262,887	189,217	73,670	41,332	3.6/214	1.0/8	1.1/38	0/0/0/0/0	C +NC 3.0
[R27-la] WFL	36,417	32,414	4,003	121	9.1/524	1.7/6	4.3/236	46/29/21/0/4	C +NC 4.0

Table 4. Language resources harmonised in the UDer collection and their basic quantitative features. Columns Tree size, Tree depth, and Tree out-degree are presented in average / maximum value format. Part-of-speech distribution is ordered as follows: nouns, adjectives, verbs, adverbs, and other categories. (C is abbreviation for CC BY-SA in License)

and named entities) occur in E-CELEX (6,600), FinnWordNet (1,297), the Morpho-Semantic Database (105), DerivBase.Ru (60), EstWordNet (14), DerIvaTario (6), and Démonette (2).

Rel(ations). Table 4 counts only relations involved in tree-shaped families; non-tree relations (compounding, etc.) are not included, although they are present in the harmonised data too. Compounds are captured and connected to their base lexemes in D-CELEX (3,949), G-CELEX (2,563), Word Formation Latin (1,747), E-CELEX (621), and DeriNet (600; other 32,479 compound lexemes are identified by a label without (yet) being connected to their base lexemes).

Fam(ilies) and Singl(etons). The column of derivational families counts only families which have more than one lexeme, including families created by splitting the original families into more parts during the identification of rooted trees. Links between the new roots of the divided families are also stored in the harmonised data. As for the amount of singletons (one-node trees), it seems to correlate with the ways

the original resources have been created. Many singleton trees occur in resources that were built by finding derivational relations within the lexeme set (bottom-up approach), i.e., the CELEXes, DeriNet, DeriNet.ES, DErivBase, and the Polish Word-Formation Network, whereas the lower number of singletons is documented for resources that included lexemes depending on whether the lexeme was derivationally linked to any other lexeme (top-down approach). The number of singleton trees could increase during the harmonisation as a result of splitting the original families.

Tree size, depth, and out-deg(ree). The columns describe the average and maximum size of derivational families, their average and maximum depth (i.e., the distance from the tree root to the furthest node), and out-degree (i.e., the highest number of direct children of a single node). In average, the biggest derivational families can be found in EtymWordNet-pol,[9] Word Formation Latin, DeriNet.FA, and DeriNet, while the smallest families are in the CELEXes and DErivBase, as their data is made up mostly of singletons; a similar tendency can also be seen for the maximum tree sizes. The biggest tree with more than 1.6 thousand lexemes is captured in Derinet, namely for the Czech verb *dát* '*to give*'. The tree depths and out-degrees document that NomLex-PT and the Morpho-Semantic Database contain nouns derived from verbs only. The small absolute depths combined with high absolute out-degrees indicate that derivational families are relatively spread in Etymological WordNets.

POS distrib(ution). Lexemes are assigned part-of-speech tags only in less than a half of the harmonised resources. Nouns, adjectives, verbs, and adverbs are captured in CatVar, the CELEXes, DeriNet, DerivBase.Ru, and EstWordNet. Démonette, DErivBase, DerivBase.HR, and FinnWordNet lack adverbs whereas Démonette and DErivBase have a low number of adjectives. Word Formation Latin contains pronouns, auxiliaries, and lexemes unspecified for the part of speech. The Morpho-Semantic Database and NomLex-PT are limited to nouns and verbs only.

Semantic labels. Derivational meanings are labelled in Démonette, DeriNet, and the Morpho-Semantic Database. However, the labels cannot be merged as they have different interpretations; harmonisation of these labels is one of the future tasks. While the Morpho-Semantic Database assigns labels based on WordNet semantic types, i.e., Agent, Body, By, Destination, Event, Instrument, Location, Material, Property, Result, State, Undergoer, Uses, and Vehicle; morpho-syntactic tags are used in Démonette, i.e., ACT, RES, AGF, AGM, and PRP; and labels in DeriNet are rooted in comparative semantic concepts, namely DIMINUTIVE, POSSESSIVE, FEMALE, ITERATIVE (inspired by Bagasheva 2017), and ASPECT.

Morphological segmentation. A partial or complete morphological segmentation is included in the CELEXes, Démonette, DeriNet, DerIvaTario, DerivBase, DerivBase.Ru, and Word Formation Latin, though the scope of annotation differs largely. While a complete morphological segmentation occurs in the CELEXes and DerIvaTario,

[9] It should be mentioned that Etymological WordNets often represent inflectionally related lexemes as derivation.

only those morphemes which are part of a particular derivational relation are segmented in Démonette and Word Formation Latin. Morhological segmentation in DeriNet is currently limited to identification of root morphemes.

4.2. Publishing and licensing

The presented collection UDer 1.0 is freely available in a single data package in the LINDAT/CLARIAH CZ repository[10] under the licenses listed in Table 4. Relevant scripts for harmonising the original resources and releasing the UDer collection are available in the GitHub repository.[11] The UDer data can be processed using software developed within the DeriNet project, especially the DeriSearch tool,[12] and the Python application interface for DeriNet 2.0;[13] see UDer web page for more information and updates.[14]

4.3. Query interface

As illustrated in Figure 8, UDer trees can grow quite big. Even if the file format is text-based, it has been optimised rather for data exchange, and it is difficult to read by a naked eye (especially when patterns composed of multiple nodes are considered). Thus a specialised interface is needed for human users to browse the UDer data.

Currently we use an updated version of DeriSearch (Vidra and Žabokrtský, 2017) for searching and visualisation purposes. The query language of the tool was recently extended to support querying non-tree graph structures such as compounding, as well as specific node and relation features such as morphological segmentation and semantic labels (Vidra and Žabokrtský, 2020).

The query language supports searching for individual lexemes by imposing regular expression conditions (possibly more of them, combined using logical operators) on their properties. At the same time, it supports searching for contiguous subgraphs composed of multiple lexemes connected by word-formation relations.

Figure 9 shows results for two sample queries. The first query searches for a single specific node whose lemma is *betreffen*, while the second query expresses a more general pattern that matches any node which is a verb and from which at least three child nodes are derived (without conditioning their properties).

Quantitative results of another set of DeriSearch queries, this time applied across several languages, are summarised in Table 5. Given that POS is the only node attribute conditioned in the four queries, we evaluated the queries on all UDer datasets

[10] http://hdl.handle.net/11234/1-3236

[11] https://github.com/lukyjanek/universal-derivations

[12] https://quest.ms.mff.cuni.cz/derisearch2/v2/databases/

[13] https://github.com/vidraj/derinet/tree/master/tools/data-api/derinet2

[14] http://ufal.cz/universal-derivations

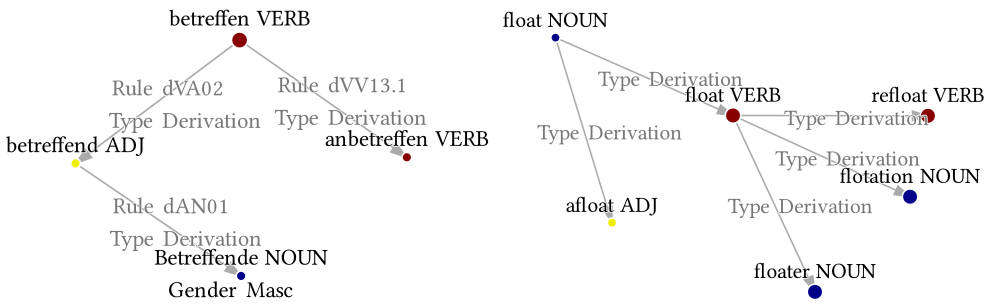


Figure 9. Result of searching for `[lemma="betreffen"]` in DErivBase using DeriSearch (left). Notice that the noun Betreff *is not present, as explained in Section 3.4 and illustrated in Figure 6. One of the results for `[pos="VERB"] ([], [], [])` in the E-CELEX database, visualised by DeriSearch (right).*

in which POS values are available. Please note that the columns are described using a shortened notation, for instance `V(N,N,N)` corresponds to query:

```
[pos="VERB"]([pos="NOUN"],[pos="NOUN"],[pos="NOUN"])
```

Let us use, for example, French, German, Croatian, and Czech to illustrate the subgraphs found by DeriSearch:

- `V(N,N,N)` represents three different nouns derived from the same verb, such as in the case of the French nouns *armeteur* '*armeteer*', *armeur* '*armorer*', and *armement* '*armament*', all derived from the verb *armer* '*to arm*'.
- `V(A,A)` represents two adjectives derived from the same verb, such as the German adjectives *heimatlich* '*native*' and *heimatlos* '*without homeland*' derived from *Heimat* '*homeland*'.
- `V N A` represents patterns in which an adjective is derived from a noun which is derived from a verb, such as in the case of Croatian triple *obujmiti* '*to embrace*', *obujam* '*volume*', *obujamski* '*volumetric*'.
- `N(A,A)` represents two adjectives derived from the same noun, such as *Aristotelův* '*Aristotle's*' and *aristotelský* '*Aristotelian*' derived from *Aristoteles* '*Aristotle*' in Czech.

When comparing individual lines in Table 5, one quickly notices striking quantitative discrepancies among the resources. The counts seem to be far from correlated, and hence the variability can be hardly attributed only to different sizes of the input resources (though their sizes differ in the order of magnitude). The existence of genuine linguistic differences among the languages cannot serve as a sole explanation either, as resources for related languages (or even two resources for a same language) differ substantially too. The most viable explanation is that—is spite of our harmon-

	V(N,N,N)	V(A,A)	V N A	N(A,A)
[R1-en] CatVar	558	863	508	156
[R2-nl] D-CELEX	3	0	1	326
[R3-en] E-CELEX	96	49	125	50
[R4-de] G-CELEX	160	123	273	166
[R5-fr] Démonette	1664	0	408	2
[R6-cs] DeriNet	1510	54874	3655	9124
[R9-it] DerIvaTario	6	7	11	1
[R10-de] DErivBase	332	1363	369	283
[R11-hr] DerivBase.Hr	86	445	385	1062
[R12-ru] DerivBase.Ru	1166	265	2342	2511
[R13-et] EstWordNet	0	0	0	0
[R23-fi] FinnWordNet	559	79	263	634
[R24-pt] NomLex-PT	303	0	0	0
[R25-en] The M-S Database	192	0	0	0
[R27-la] WFL	1010	654	468	995

Table 5. Counts of results found for four sample queries (shortened notation) across different resources.

isation efforts—there is still a long way to overcome the diversity of design decisions petrified in the original data resources and to reach fully comparable networks.

5. Conclusions

This paper presented a procedure for unifying annotation schemas of resources capturing word-formation. Twenty-seven resources covering 20 (mostly European) languages were harmonised using a semiautomatic procedure, and their harmonised versions were publicly released under the title Universal Derivations (UDer 1.0). The harmonised resources allow processing data of multiple languages by the same software tools. The DeriSearch engine was presented here as a tool for visualisation and querying the data.

One of the goals of our harmonisation efforts is to initiate a discussion about the design decisions made, including the choice of the target schema and particular features to harmonise. Harmonisation is necessarily a compromise in that it is impossible to keep all information and allow processing it in an efficient unified way at the same time. Still, we hope that the benefits of the presented efforts outweigh the negatives.

Acknowledgements

We would like to thank all researchers who made their derivational resources publicly available under open licenses. Special thanks also go to Anna Nedoluzhko for manual annotations of Russian data and for valuable comments on the draft of this article.

This work was supported by the Grant No. GA19-14534S of the Czech Science Foundation, by the Charles University Grant Agency (project No. 1176219) and by the

SVV project number 260 575. It has been using language resources developed, stored, and distributed by the LINDAT/CLARIAH CZ project (LM2015071, LM2018101).

Bibliography

Ansari, Ebrahim, Zdeněk Žabokrtský, Hamid Haghdoost, and Mahshid Nikravesh. Persian Morphologically Segmented Lexicon 0.5, 2019. URL `http://hdl.handle.net/11234/1-3011`. LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Baayen, Harald R., Richard Piepenbrock, and Leon Gulikers. CELEX2, 1995. Linguistic Data Consortium, Catalogue No. LDC96L14.

Bagasheva, Alexandra. Comparative Semantic Concepts in Affixation. In Santana-Lario, Juan and Salvador Valera, editors, *Competing Patterns in English Affixation*, pages 33–65. Peter Lang, Bern, 2017. ISBN 978-30-3432-701-5.

Buchholz, Sabine and Erwin Marsi. CoNLL-X Shared Task on Multilingual Dependency Parsing. In *Proceedings of the 10th Conference on Computational Natural Language Learning*, pages 149–164. ACL, 2006. doi: 10.3115/1596276.1596305.

Chu, Yoeng-Jin and T. H. Liu. On the Shortest Arborescence of a Directed Graph. *Scientia Sinica*, 14:1396–1400, 1965.

De Paiva, Valeria, Livy Real, Alexandre Rademaker, and Gerard de Melo. NomLex-PT: A Lexicon of Portuguese Nominalizations. In *Proceedings of the 9th International Conference on Language Resources and Evaluation*, pages 2851–2858. ELRA, 2014.

Dokulil, Miloš. *Tvoření slov v češtině 1: Teorie odvozování slov*. Academia, Prague, 1962.

Edmonds, Jack. Optimum Branchings. *Journal of Research of the national Bureau of Standards*, 71B (4):233–240, 1967. doi: 10.6028/jres.071B.032.

Faryad, Ján. Identifikace derivačních vztahů ve španělštině. Technical Report TR-2019-63, Faculty of Mathematics and Physics, Charles University, 2019.

Fellbaum, Christiane, Anne Osherson, and Peter E Clark. Putting Semantics into WordNet's "Morphosemantic" Links. In *Language and Technology Conference*, pages 350–358. Springer, 2007. doi: 10.1007/978-3-642-04235-5_30.

Gerard, de Melo. Etymological Wordnet: Tracing The History of Words. In *Proceedings of the Language Resources and Evaluation (LREC-2014)*, pages 1148–1154, Reykjavik, Iceland, 5 2014. ELRA. ISBN 978-2-9517408-8-4.

Glare, P. G. W. *Oxford Latin dictionary*. Clarendon Press, Oxford, 1968.

Habash, Nizar and Bonnie Dorr. A Categorial Variation Database for English. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, volume 1, pages 17–23, Stroudsburg, 2003. Association for Computational Linguistics. doi: 10.3115/1073445.1073458.

Haghdoost, Hamid, Ebrahim Ansari, Zdeněk Žabokrtský, and Mahshid Nikravesh. Building a Morphological Network for Persian on Top of a Morpheme-Segmented Lexicon. In *Proceedings of the 2nd Workshop on Resources and Tools for Derivational Morphology*. Charles University, 2019.

Hathout, Nabil. Morphonette: A Morphological Network of French. *arXiv preprint arXiv:1005.3902*, 2010.

Hathout, Nabil and Fiammetta Namer. Démonette, a French Derivational Morpho-Semantic Network. *Linguistic Issues in Language Technology*, 11:125–162, 2014.

Iacobini, Claudio. Base and Direction of Derivation. In *Morphology. An International Handbook on Inflection and Word-formation*, volume 1, pages 865–876. Mouton de Gruyter, 2000.

Jaccard, Paul. The Distribution of the Flora in the Alpine Zone. *New Phytologist*, 11(2):37–50, 1912. doi: 10.1111/j.1469-8137.1912.tb05611.x.

Jaro, Matthew A. Advances in Record-linkage Methodology as Applied to Matching the 1985 Census of Tampa, Florida. *Journal of the American Statistical Association*, 84(406):414–420, 1989. doi: 10.1080/01621459.1989.10478785.

Kahusk, Neeme, Kadri Kerner, and Kadri Vider. Enriching Estonian WordNet with Derivations and Semantic Relations. In *Baltic hlt*, pages 195–200, 2010.

Kerner, Kadri, Heili Orav, and Sirli Parm. Growth and Revision of Estonian WordNet. In *Principles, Construction and Application of Multilingual WordNets*, pages 198–202. Narosa Publishing House, 2010.

Kyjánek, Lukáš, Zdeněk Žabokrtský, Jonáš Vidra, and Magda Ševčíková. Universal Derivations v0.5, 2019. URL `http://hdl.handle.net/11234/1-3041`. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Kyjánek, Lukáš, Zdeněk Žabokrtský, Jonáš Vidra, and Magda Ševčíková. Universal Derivations v1.0, 2020. URL `http://hdl.handle.net/11234/1-3236`. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Kyjánek, Lukáš. Morphological Resources of Derivational Word-Formation Relations. Technical Report TR-2018-61, Faculty of Mathematics and Physics, Charles University, 2018.

Kyjánek, Lukáš. Harmonisation of Language Resources for Word-Formation of Multiple Languages. Master's thesis, Charles University, Faculty of Mathematics and Physics, 2020.

Kyjánek, Lukáš, Zdeněk Žabokrtský, Magda Ševčíková, and Jonáš Vidra. Universal Derivations Kickoff: A Collection of Harmonized Derivational Resources for Eleven Languages. In *Proceedings of the Second International Workshop on Resources and Tools for Derivational Morphology*, pages 101–110, Prague, 2019. ISBN 978-80-88132-08-0.

Lango, Mateusz, Magda Ševčíková, and Zdeněk Žabokrtský. Semi-Automatic Construction of Word-Formation Networks (for Polish and Spanish). In *Proceedings of the 11th International Conference on Language Resources and Evaluation*, pages 1853–1860. ELRA, 2018.

Levenshtein, Vladimir I. Binary Codes Capable of Correcting Deletions, Insertions and Reversals. *Soviet Physics Doklady*, 10(8):707–710, 1966.

Lieber, Rochelle and Pavol Štekauer. *The Oxford handbook of derivational morphology*. Oxford University Press, Oxford, 2014. doi: 10.1093/oxfordhb/9780199641642.001.0001.

Lindén, Krister and Lauri Carlson. FinnWordNet–Finnish WordNet by Translation. *LexicoNordica – Nordic Journal of Lexicography*, 17:119–140, 2010.

Lindén, Krister, Jyrki Niemi, and Mirka Hyvärinen. Extending and updating the Finnish Wordnet. In *Shall We Play the Festschrift Game?*, pages 67–98. Springer, 2012. doi: 10.1007/978-3-642-30773-7_7.

Litta, Eleonora, Marco Passarotti, and Chris Culy. Formatio Formosa est. Building a Word Formation Lexicon for Latin. In *Proceedings of the 3rd Italian Conference on Computational Linguistics*, pages 185–189, 2016. doi: 10.4000/books.aaccademia.1799.

Matoušek, Jiří and Jaroslav Nešetřil. *Invitation to Discrete Mathematics*. Oxford University Press, Oxford, 2009. ISBN 978-0-19-857043-1.

Maziarz, Marek, Maciej Piasecki, Ewa Rudnicka, Stan Szpakowicz, and Paweł Kędzia. plWordNet 3.0 – a Comprehensive Lexical-Semantic Resource. In *Proceedings of the 26th International Conference on Computational Linguistics*, pages 2259–2268, Osaka, Japan, December 2016. The COLING 2016 Organizing Committee.

McDonald, Ryan, Joakim Nivre, Yvonne Quirmbach-Brundage, Yoav Goldberg, Dipanjan Das, Kuzman Ganchev, Keith Hall, Slav Petrov, Hao Zhang, Täckström Oscar, Bedini Claudia, Castelló B. Núria, and Lee Jungmee. Universal Dependency Annotation for Multilingual Parsing. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, volume 2, pages 92–97. ACL, 2013.

Miller, George. *WordNet: An Electronic Lexical Database*. MIT press, 1998. ISBN 9780262561167.

Namer, Fiammetta. Automatiser l'analyse morpho-sémantique non affixale: le système DériF. *Cahiers de grammaire*, 28:31–48, 2003.

Nivre, Joakim, Marie Catherine De Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajič, Christopher D Manning, Ryan McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, Tsarfaty Reut, and Zeman Daniel. Universal Dependencies v1: A Multilingual Treebank Collection. In *Proceedings of the Language Resources and Evaluation (LREC-2016)*, pages 1659–1666, Portorož, Slovenia, 2016a. ELRA. ISBN 978-2-9517408-9-1.

Nivre, Joakim, Marie-Catherine De Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajič, Christopher D Manning, Ryan McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, Tsarfaty Reut, and Zeman Daniel. Universal Dependencies v1: A Multilingual Treebank Collection. In *Proceedings of the 10th International Conference on Language Resources and Evaluation*, pages 1659–1666. ELRA, 2016b.

Shafaei, Elnaz, Diego Frassinelli, Gabriella Lapesa, and Sebastian Padó. DErivCELEX: Development and Evaluation of a German Derivational Morphology Lexicon based on CELEX. In *Proceedings of the Workshop on Resources and Tools for Derivational Morphology*, pages 117–127, Milan, Italy, 2017. ISBN 978-88-9335-225-3.

Šnajder, Jan. DerivBase.hr: A High-Coverage Derivational Morphology Resource for Croatian. In *Proceedings of the 9th International Conference on Language Resources and Evaluation*, pages 3371–3377. ELRA, 2014.

Talamo, Luigi, Chiara Celata, and Pier Marco Bertinetto. DerIvaTario: An Annotated Lexicon of Italian Derivatives. *Word Structure*, 9(1):72–102, 2016. doi: 10.3366/word.2016.0087.

Tanguy, Ludovic and Nabil Hathout. Webaffix: un outil d'acquisition morphologique dérivationnelle à partir du Web. In *Actes de la 9e Conférence Annuelle sur le Traitement Automatique des Langues Naturelles (TALN-2002)*, Nancy, France, 2002. ATALA.

Vidra, Jonáš, Zdeněk Žabokrtský, Lukáš Kyjánek, Magda Ševčíková, and Šárka Dohnalová. DeriNet 2.0, 2019a. URL http://hdl.handle.net/11234/1-2995. LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Vidra, Jonáš, Zdeněk Žabokrtský, Magda Ševčíková, and Lukáš Kyjánek. DeriNet 2.0: Towards an All-in-One Word-Formation Resource. In *Proceedings of the 2nd Workshop on Resources and Tools for Derivational Morphology*. Charles University, 2019b.

Vidra, Jonáš and Zdeněk Žabokrtský. Online Software Components for Accessing Derivational Networks. In *Proceedings of the Workshop on Resources and Tools for Derivational Morphology*, pages 129–139. EDUCatt, 2017.

Vidra, Jonáš and Zdeněk Žabokrtský. Next step in online querying and visualization of word-formation networks. In *Proceedings of the 23rd International Conference on Text, Speech and Dialogue* (*TSD 2020*). Springer, 2020. doi: 10.1007/978-3-030-58323-1_15. In print.

Vodolazsky, Daniil. DerivBase.Ru: A Derivational Morphology Resource for Russian. In *Proceedings of the Language Resources and Evaluation* (*LREC-2020*), volume 20, pages 3930–3936, Marseille, France, 2020.

Winkler, William E. String Comparator Metrics and Enhanced Decision Rules in the Fellegi-Sunter Model of Record Linkage. In *Proceedings of the Section on Survey Research*, pages 354–359. ERIC, 1990.

Zeller, Britta, Jan Šnajder, and Sebastian Padó. DErivBase: Inducing and Evaluating a Derivational Morphology Resource for German. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, volume 1, pages 1201–1211. ACL, 2013.

Zeller, Britta, Sebastian Padó, and Jan Šnajder. Towards Semantic Validation of a Derivational Lexicon. In *Proceedings of COLING 2014*, pages 1728–1739, Dublin, Ireland, 8 2014. Dublin City University and Association for Computational Linguistics. ISBN 978-1-941643-26-6.

Zeman, Daniel, Ondřej Dušek, David Mareček, Martin Popel, Loganathan Ramasamy, Jan Štěpánek, Zdeněk Žabokrtský, and Jan Hajič. HamleDT: Harmonized Multi-Language Dependency Treebank. *Language Resources and Evaluation*, 48(4):601–637, 2014. doi: 10.1007/s10579-014-9275-2.

Address for correspondence:
Lukáš Kyjánek
kyjanek@ufal.mff.cuni.cz
Institute of Formal and Applied Linguistics
Faculty of Mathematics and Physics
Charles University
Malostranské náměstí 25
118 00 Praha 1, Czech Republic

PBML

The Prague Bulletin of Mathematical Linguistics
NUMBER 115 OCTOBER 2020 31–50

Inferring Highly-dense Representations for Clustering Broadcast Media Content

Esaú Villatoro-Tello,[a b] Shantipriya Parida,[b] Petr Motlicek,[b] Ondřej Bojar[c]

[a] Universidad Autónoma Metropolitana, Unidad Cuajimalpa, Mexico City, Mexico.
[b] Idiap Research Institute, Rue Marconi 19, 1920 Martigny, Switzerland.
[c] Charles University, Faculty of Mathematics and Physics, Institute of Formal and Applied Linguistics
Malostranské náměstí 25 118 00 Praha 1, Czech Republic

Abstract

We propose to employ a low-resolution representation for accurately categorizing spoken documents. Our proposed approach guarantees document clusters using a highly dense representation. Performed experiments, using a dataset from a German TV channel, demonstrate that using low-resolution concepts for representing the broadcast media content allows obtaining a relative improvement of 70.4% in terms of the Silhouette coefficient compared to deep neural architectures.

1. Introduction

Current broadcast platforms utilize the Internet as a cross-promotion source, thus, their produced materials tend to be very short and thematically diverse. Besides, modern Web technologies allow the rapid distribution of these informative content through several platforms. As a result, the broadcast media content monitoring represents a challenging scenario for current Natural Language Understanding (NLU) approaches to efficiently exploit this type of data due to a lack of structuring and reliable information associated with these contents (Morchid and Linarès, 2013; Doulaty et al., 2016; Staykovski et al., 2019). Furthermore, if we consider that documents are very short and that they come from a very narrow domain, the task of clustering becomes harder.

Traditionally, the Bag-of-Words (BoW) has been the most widely used text representation technique for solving many text-related tasks, including document cluster-

 Corresponding author: shantipriya.parida@idiap.ch
Cite as: Esaú Villatoro-Tello, Shantipriya Parida, Petr Motlicek, Ondřej Bojar. Inferring Highly-dense Representations for Clustering Broadcast Media Content. The Prague Bulletin of Mathematical Linguistics No. 115, 2020, pp. 31–50. doi: 10.14712/00326585.004.

ing, due to its simplicity and efficiency (Ribeiro-Neto and Baeza-Yates, 1999). However, the BoW has two major drawbacks: *i*) document representation is generated in a very high-dimensional space, *ii*) it is not feasible to determine the semantic similarity between words. As widely known, previous problems increase when documents are short texts (Li et al., 2016). It becomes more difficult to statistically evaluate the relevance of words given that most of the words have low-frequency occurrences, the BoW representation from short-texts results in a higher sparse vector, and the distance between similar documents is not very different than the distance between more dissimilar documents.

To overcome some of the BoW deficiencies, semantic analysis (SA) techniques attempt to interpret the meaning of the words and text fragments by calculating their relationship with a set of predefined concepts or topics (Li et al., 2011). Examples of SA techniques are LDA (Blei et al., 2003), LSA (Deerwester et al., 1990), and word embeddings (Le and Mikolov, 2014; Bojanowski et al., 2017; Devlin et al., 2019). Accordingly, these strategies learn word or document representations based on the combination of the underlying semantics in a dataset. Similarly, more recent approaches, with the help of word embeddings, learn text representations using deep neural network architectures for document classification (De Boom et al., 2016; Adhikari et al., 2019; Ostendorff et al., 2019; Sheri et al., 2019). However, most of these approaches focus either on solving supervised classification tasks or clustering formal-written short documents.

In this paper, we propose an efficient technological solution for the unsupervised categorization of broadcast media content, i.e., spoken documents. Our proposed approach generates document clusters using a highly dense representation, referred to as low-resolution concepts. We first identify the fundamental semantic elements (i.e., concepts) in the document collection, then, these are used to build the low-resolution representation, which is later used in an unsupervised categorization process. One major advantage of our proposed approach is it's easy to interpret, explicit, and profound representation, allowing the end-users understanding of document vectors and their differences.

The main contributions of this paper are summarized as follows: *i*) To the best of our knowledge, this is the first attempt to explore the feasibility and effectiveness of the low-resolution bag-of-concepts in solving one particular unsupervised task, broadcast media content categorization; *ii*) We conducted our experiments on a real-life dataset of German spoken documents, achieving good performance in terms of three internal evaluation metrics, allowing our method to be considered for practical deployment; *iii*) We evaluate the performance of our proposed method in three well-known datasets (formal written documents).

The remainder of the paper is organized as follows: a brief description of the related work is given in Section 2, in Section 3 we describe the proposed methodology, Section 4 we provide some details regarding the employed dataset. Experimental re-

sults and analyses are presented in Sections 5 and 6. Finally, in Section 7 we draw our main conclusions and future work directions.

2. Related Work

Our work is mainly related to topic modeling or topic discovery. As known, topic discovery aims to use statistical information of word occurrences to obtain the underlying semantics contained in a document set. The most popular textual topic modelling are based on Latent Dirichlet Allocation (LDA) (Blei et al., 2003), Bayesian methods represented by latent semantic analysis (LSA) (Deerwester et al., 1990), Hierarchical Dirichlet Process (HDP) (Teh et al., 2005).

During recent years, models based on deep neural networks have emerged as a viable alternative for topic discovery. For example, the replicated softmax model (RSM), based on Restricted Boltzmann Machines (Hinton and Salakhutdinov, 2009), which is capable to estimate the probability of observing a new word in a document given previously observed words, thus RSM can learn efficient document representations. More recently, Variational Autoencoders (VAEs) have been successfully adapted for text topic modeling. The Neural Variational Document Model (NVDM) (Miao et al., 2016) for text modeling is an extension of a standard VAE, with an encoder that learns Gaussian distribution and a softmax decoder capable of reconstructing documents in a semantic word embedding space. In (Silveira et al., 2018) authors propose a VAE-based on Gumbel-Softmax (GSDTM) and Logistic-normal Mixture (LMDTM) for text topic modelling. In (Wang et al., 2020) authors propose a neural topic modeling approach, called Bidirectional Adversarial Topic (BAT) model, which builds a two-way projection between the document-topic distribution and the document-word distribution. Although these recent approaches have demonstrated great improvement in text clustering tasks using the topic information, they all have one major disadvantage, they require great amounts of data to infer accurate semantic representations, plus the lack of interpretability.

Despite the extensive exploration of this research field, scarce work has been done to evaluate the impact of these technologies in speech-documents, i.e., textual transcriptions obtained from speech. Contrary to formal documents, textual transcript represents a more challenging scenario as they represent very short documents, containing several speech phenomena such as hesitation, fillers, repetition, etc. Accordingly, in this paper, we evaluate the impact of several clustering strategies for broadcast media categorization. Our proposed approach generates document clusters using highly dense representation, which are easy to interpret by a human judge. The recent relevant work to ours is proposed by (Kim et al., 2017), which proposes a bag-of-concepts approach to generate alternative document representations to overcome the lack of interpretability of word2vec and doc2vec methodologies. However, contrary to this particular work, our method is particularly suited for very short spoken documents (transcripts), and we use highly dense representations, i.e., a very small

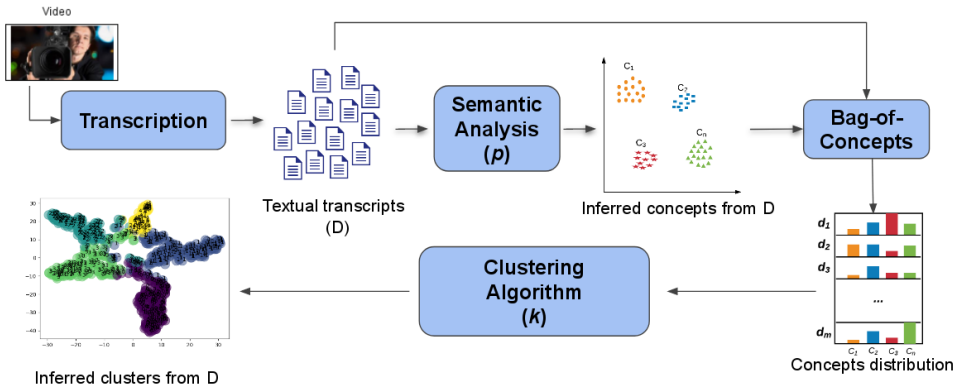


Figure 1: General framework to categorize spoken-documents using low-resolution concepts.

set of features is used to represent the concepts contained in the dataset. We evaluate our proposed method in a real-life dataset extracted to form a German tv channel and we also evaluate our method's performance in three benchmark corpora.

3. Proposed Method

Inspired by the work of (Kim et al., 2017; López-Monroy et al., 2018), we propose using a highly dense representation, denominated low-resolution concepts, for solving the task of clustering short transcript-texts, i.e., broadcast media documents. The intuition behind this approach is that highly abstract semantic elements (concepts) are good discriminators for clustering very short transcript texts that come from a narrow domain. The proposed methodology is depicted in Figure 1. Generally speaking, we first identify the underlying concepts contained in the dataset. For this, we can employ any semantic analysis (SA) approach for learning words representation; thus, learned representation allows us to generate sets of semantically associated words. After obtaining the main concepts, documents are represented by a condensed vector, which counts for the occurrences of the concepts, i.e., a concept distribution vector. Finally, the build texts representation serves as the input to a clustering process, in this case, the K-means algorithm.

More formally, let $\mathcal{D} = \{d_1, d_2, \ldots, d_n\}$ denote the set of short transcript texts, and let $\mathcal{V} = \{w_1, w_2, \ldots, w_m\}$ represent the vocabulary of the document collection $\mathcal{D}$. As first step, we aim at inferring the underlying set of concepts $\mathcal{C} = \{c_1, c_2, \ldots, c_p\}$ contained in $\mathcal{D}$, where every $c_l \in \mathcal{C}$ is a set formed by semantically related words. Notice that in order to obtain the concepts $\mathcal{C}$ we can apply any SA technique for learning the vector representation $\boldsymbol{v_i}$ of each word $w_i \in \mathcal{V}$, for example LDA, LSA, or word

embeddings. Next, for obtaining the document $\mathbf{d}_j$ representation, we account for the occurrence of each c_l within d_i, in other words, the document vector $\mathbf{d}_j$ is a vector that contains concepts distribution. Finally, the generated document-concepts matrix $\mathbf{M}_{\mathcal{D}\times\mathcal{C}}$ serves as the input to a clustering process aiming at finding the more suitable documents groups according to the concept-based representation. Henceforth, we will refer to the document-concepts matrix as the Bag-of-Concepts (BoC) representation.

The proposed method has two main parameters, the resolution parameter (p) and the group parameter (k). The former, p, represents the number of concepts that will be generated from the SA step. The lower the number of concepts, the more abstract the resolution. The second parameter, k, indicates the number of categories to be generated from the clustering process. Given the nature of the dataset, i.e., very short texts from a narrow domain, we hypothesize that the clustering algorithm will be able to find groups of documents that share the same amount of information about the same sub-set of concepts, resulting in a more coherent categorization of the documents. Thus, using low-resolution concepts will generate groups of documents referring to the same general topics, while using higher resolution values will result in a more fine-grained topic categorization of the documents.

4. Dataset Description

The dataset used in our paper is from n-tv[1], a German free-to-air television news channel. There are mainly two different sets of files in the proprietary data. One part of the dataset is represented by the speech segments (audio data) with an average duration of 1.5 minutes where each recording has multiple speakers recorded in a relatively noisy environment. The other part of the dataset is the textual transcripts (German) associated with the speech segments. Each of the transcript files represents an article (short text documents), which usually are spread across different topics. See for example a small fragment of an article shown in Table 1. This example, when given to experts, is categorized as 'politics' and as an 'economy' article, which is somehow correct given that both topics are present in the article. This occurs repeatedly across articles due to the interviewed people often mix topics when spontaneously speaking, making the categorization task even more challenging.

For our experiments, the employed dataset comprises a total of 697 articles. Table 2 shows some statistics from the employed dataset; before applying any pre-processing operation and after pre-processing. As pre-processing operations, we removed stop-words, numbers, special symbols, all the words are converted to lower-case. [2]We compute the average number of tokens, vocabulary, and lexical richness (LR) in the dataset. A couple of main observations can be done at this point. On the one hand,

[1]https://www.n-tv.de/

[2]We did not make any special processing for German compounds words.

Original German fragment
Arbeitsminister Hubertus Heil **kämpft** für **befristete Teilzeit**. Also dafür dass man nicht nur von Voll-zur Teilzeit sondern eben auch wieder zurück wechseln kann ...der **Arbeitgeber** darf den Antrag auf Teilzeit auch nicht einfach so ausschlagen außer es gibt betriebliche Gründe... bei **Unternehmen** mit mehr als 200 **Mitarbeitern** habe alle ein Recht auf befristete Teilzeit...zudem kann der Arbeitgeber den Antrag auf befristete Arbeitszeit ablehnen wenn diese ein Jahr unter- oder fünf Jahre überschreitet.
Closest English translation
Minister of Labor Hubertus Heil is **fighting** for **part-time work**. So that you can not only switch from full-time to part-time but also back again ... the **employer** may not simply refuse the application for part-time unless there are operational reasons ... in **companies** with more than 200 **employees**, everyone has a right to temporary part-time work ... the employer can also reject the application for limited working time if it exceeds one year less than or five years.

Table 1: Extracted fragment from the n-tv dataset. Letters in **bold** represent keywords associated with *politic* and *economic* topics.

we notice that individual texts are very short, on average 63.02 tokens with an average vocabulary of 47.86 words, resulting in a very high LR (0.785). This suggests that very few words are repeated within one article, very few redundancies, which represents a challenge for frequency-based methods. On the other hand, globally speaking, the complete dataset has an LR=0.272, which indicates, to some extent, that the information across texts is highly overlapped (narrow domains).

4.1. Benchmark datasets

To validate our proposal, we also evaluate our method in the following three benchmark datasets:

- **AG's news corpus.** We used the as employed in (Zhang et al., 2015). It contains categorized news articles (4 classes) from more than 2000 news sources. In total, this dataset contains 120000 documents in the train partition and 7600 in the test partition.
- **Reuters.** These documents appeared on the Reuters newswire in 1987 and were manually classified by personnel from Reuters Ltd. Particularly, we used for our experiments the R8 partition as provided in (Cardoso-Cachopo, 2007), i.e., 5845 documents for training, and 2189 for testing divided into eight categories.
- **10KGNAD.** This dataset, based on the One Million Posts Corpus (Schabus et al., 2017), is composed of 10273 German news articles collected from an Australian online newspaper. News is categorized into 9 different topics. The train partition contains 9245 documents, while the test partition contains 1028 documents.

	W/O Pre-processing	
	Average (σ)	Total
Tokens	234.68 ($\pm$ 124.45)	163,572
Vocabulary	161.79 ($\pm$ 51.92)	22078
LR	0.717 ($\pm$ 0.073)	0.134
	W/ Pre-processing	
	Average (σ)	Total
Tokens	63.02 ($\pm$ 31.52)	43,928
Vocabulary	47.86 ($\pm$ 16.30)	11,948
LR	0.785 ($\pm$ 0.092)	0.272

Table 2: Statistics of the n-tv dataset.

5. Experimental framework

This section describes the experimental setup. First, we describe the employed methods for learning word representations. Then, we briefly explain the evaluation metrics; and finally, we describe the approaches used for comparison purposes (baselines). For all the performed experiments we ran the k-means algorithm[3] for a range of $k = 2 \ldots 15$.

5.1. Obtaining word vectors

One crucial step of our approach is learning word representations, i.e., the semantic analysis process shown in Figure 1. For this, an important parameter is the resolution value (p), which indicates the number of concepts that will be employed for building the document-concepts matrix (BoC). Accordingly, we evaluate four different methods for inferring the set $\mathcal{C}$ ($|\mathcal{C}| = p$):

- **FastText:** Concepts are inferred from applying a clustering process over $\mathcal{V}$, using as word representation pre-trained word embeddings. We used word embeddings trained with FastText[4] (Bojanowski et al., 2017) on 2 million German Wikipedia articles. This configuration is referred as: **BoC(FstTxt)**.
- **BERT:** Similar to the previous configuration but, here we use BERT (Devlin et al., 2019), a very recent approach for getting contextualized textual representations. Thus, we feed every word in $\mathcal{V}$ to BERT and preserve the encode produced by

[3] As implemented in the scikit-learn library: `https://scikit-learn.org/stable/modules/clustering.html`

[4] `https://www.spinningbytes.com/resources/wordembeddings/`

the last hidden layer (768 units) as the word vector. Performed experiments were done using the pre-trained `bert-base-german-cased` model[5]. We refer to this configuration as **BoC(BERT)**.

- **LDA:** Latent Dirichlet Allocation (Blei et al., 2003) assumes that documents are probability distributions over latent concepts, and concepts are probability distributions over words. Thus, LDA backtracks from the document level to identify concepts that are likely to have generated the dataset. We used the Mallet's LDA implementation from Gensim[6]. After obtaining the concepts, we compute the document-concepts distribution over each d_j for generating the $\mathbf{d_j}$ representation. We refer to this experiment as **BoC(LDA)**.
- **LSA:** Latent Semantic Analysis (Deerwester et al., 1990) is a purely statistical technique that applies singular value decomposition (SVD) to the term-document matrix to identify the 'latent semantic concepts'. We employed the SVD (singular value decomposition) algorithm as implemented in sklearn[7]. Then, document-concepts representation $\mathbf{d_j}$ is obtained similarly to the LDA approach. We refer to this approach as **BoC(LSA)**.

5.2. Comparisons

We compare the proposed methodology against four different approaches:

- **BoW(*tf-idf*):** Short texts are represented using a traditional Bag-of-Words (BoW) considering a *tf-idf* weighing scheme. The top 10,000 most frequent terms are employed for generating the BoW representation. Thus, once we have the document's representation, we applied the traditional k-means algorithm.
  Avg-Emb: Every short text is represented using the average of the word embeddings which are respectively weighted with their *tf-idf* score. This strategy has been considered in previous research as a common baseline (Huang et al., 2012; Lai et al., 2015; Xu et al., 2015). We used the FastText embeddings for this experiment. Similarly to the BoW baseline, once the representation is generated, we applied the k-means algorithm to perform the clustering process.
  BERT: For this, every text is feed through BERT. As the $\mathbf{d_j}$ representation we use the values of the last hidden layer (768 units). We limit the input length to 510 tokens. After generating the BERT encoding of every document, we applied the k-means algorithm.
  CNNs: Contrary to the previous baselines, this is a specific convolutional neural network designed for clustering short texts[8]. The main idea of this method is to

[5] `https://huggingface.co/transformers/pretrained_models.html`

[6] `https://radimrehurek.com/gensim/models/wrappers/ldamallet.html`

[7] `https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.TruncatedSVD.html`

[8] As implemented in `https://github.com/zqhZY/short_text_cnn_cluster`

learn deep features representations without using any external knowledge (Xu et al., 2015).

5.3. Evaluation metrics

For validating the clustering performance we employed three internal methods (Rendón et al., 2011), namely Silhouette (s) score (Rousseeuw, 1987), Calinski-Harabasz (CH) (Caliński and Harabasz, 1974), and Davies-Bouldin (DB) (Davies and Bouldin, 1979) index. Generally speaking, these metrics propose different strategies for combining the concepts of cohesion and separation for each point in the formed clusters. The cohesion value measures how closely the points in a cluster are related among them, and the separation value indicates how well a cluster is distinguished from other clusters.

Silhouette (s) score (Rousseeuw, 1987): this metric combines the concepts of cohesion and separation for each point in the formed clusters. The cohesion value measures how closely the points in a cluster are related among them, and the separation value indicates how well a cluster is distinguished from other clusters. Thus, the s score for a point i is computed as shown in expression 1.

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}} \tag{1}$$

where $a(i)$ is the cohesion score between point i and the rest of the points belonging to the same cluster; and $b(i)$ is the separation score, which represents the minimum average distance between point i and all the other points in any other cluster, of which i is not a member. At the end, the silhouette score of the clustering process is given by the mean $s(i)$ over all points. For this particular metric possible values range between -1 and 1, where a positive result indicates a better quality in the clustering.

Calinski-Harabasz (CH) index (Caliński and Harabasz, 1974): given a dataset $\mathcal{D}$ of size n, divided into k clusters, the CH index is defined as the ratio of the between-clusters dispersion mean and the within-cluster dispersion. The CH index is computed as shown in expression 2.

$$CH = \frac{SS_B}{SS_W} \times \frac{n-1}{n-k} \tag{2}$$

where SS_W is the overall within-cluster variance, and SS_B is the overall between-cluster variance. The SS_W term represents the sum of the within the sum of squares distances of each point in the cluster from that cluster's centroid, and it will decrease as the number of clusters goes up. On the other hand, the SS_B measures the variance of all the cluster centroids from the dataset's centroid. Hence, a big SS_B value means

that all centroids from all clusters are spread out, and consequently not too close to each other. Therefore, the biggest the CH index, the better the clustering output.

Davies-Bouldin (DB) index (Davies and Bouldin, 1979): this index aims to identify sets of clusters that are compact and well separated. The DB index is defined in expression 3.

$$DB = \frac{1}{k} \sum_{i,j=1}^{k} \max_{i \neq j} \left(\frac{d(i, c_i) + d(j, c_j)}{d(c_i, c_j)} \right) \tag{3}$$

where k denotes the number of formed clusters, i and j are cluster labels, then $d(i, c_i)$ is the average distance between each point of cluster i and the centroid of that cluster c_i, this is also know as cluster diameter. Likewise, $d(c_i, c_j)$ is the distance between centroids of cluster i and j respectively. Thus, the smaller the value of the DB index, the better the clustering solution.

Finally, it is worth mentioning that for the experiments performed in the AG's news, Reuters, and 10KGNAD datasets, we evaluate all the possible configurations and baselines on the test partition. Given that these datasets are labeled, we report the obtained results in terms of accuracy (ACC).

6. Results

First, we determine the impact of the resolution parameter (p) in the clustering task. Then, we compare the proposed method using the best value of p against methods described in section 5.2.

6.1. Impact of the resolution

In Figure 2 and Figure 3 we visually show the performance of the considered concepts-inferring approaches in the clustering task, i.e., BoC(FstTxt), BoC(BERT), Boc(LDA), and BoC(LSA). Each map depicts the performance of the different methods under several resolution values $p = 5, 10, 20, 50, 100, 500, 1000$ (y-axis), and several required clusters $k = 2, \ldots, 15$ (x-axis). In all cases, the darker the red color in the heat-map the better the performance, conversely, the darker the blue color the worst the performance, and if the cells tend to be white, it means an average performance. Each row in Figure 2 and Figure 3 represents the obtained performance under a different evaluation metric, s score, CH and DB index respectively. As mentioned, the lower the value of the DB index, the better the output of the clustering process. Thus, to provide the generated maps under the third row the same interpretation, we subtract the maximum obtained value under the DB metric to each of the original results.

From these experiments we observe the following: (*1*) Using low-resolution values ($p = 5, 10$) allows us to obtain better performance, showing a consistent behavior

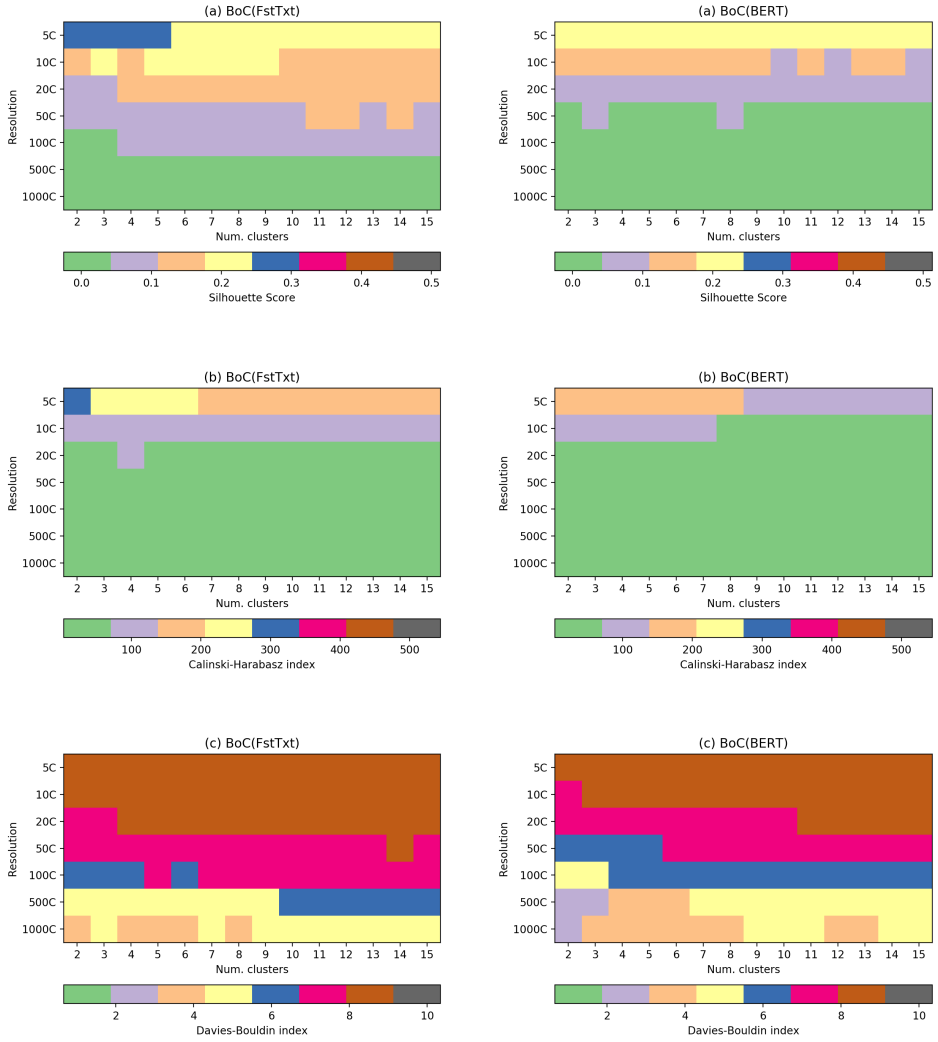


Figure 2: Heatmaps showing the impact of the resolution parameter (p) in the clustering task. First row depicts results in terms of the *s* score, second row shows the CH index, and third row represents the DB index. Graphs in the same column were generated using the same approach for inferring word representations, specifically, here we are comparing FastText and BERT approaches.

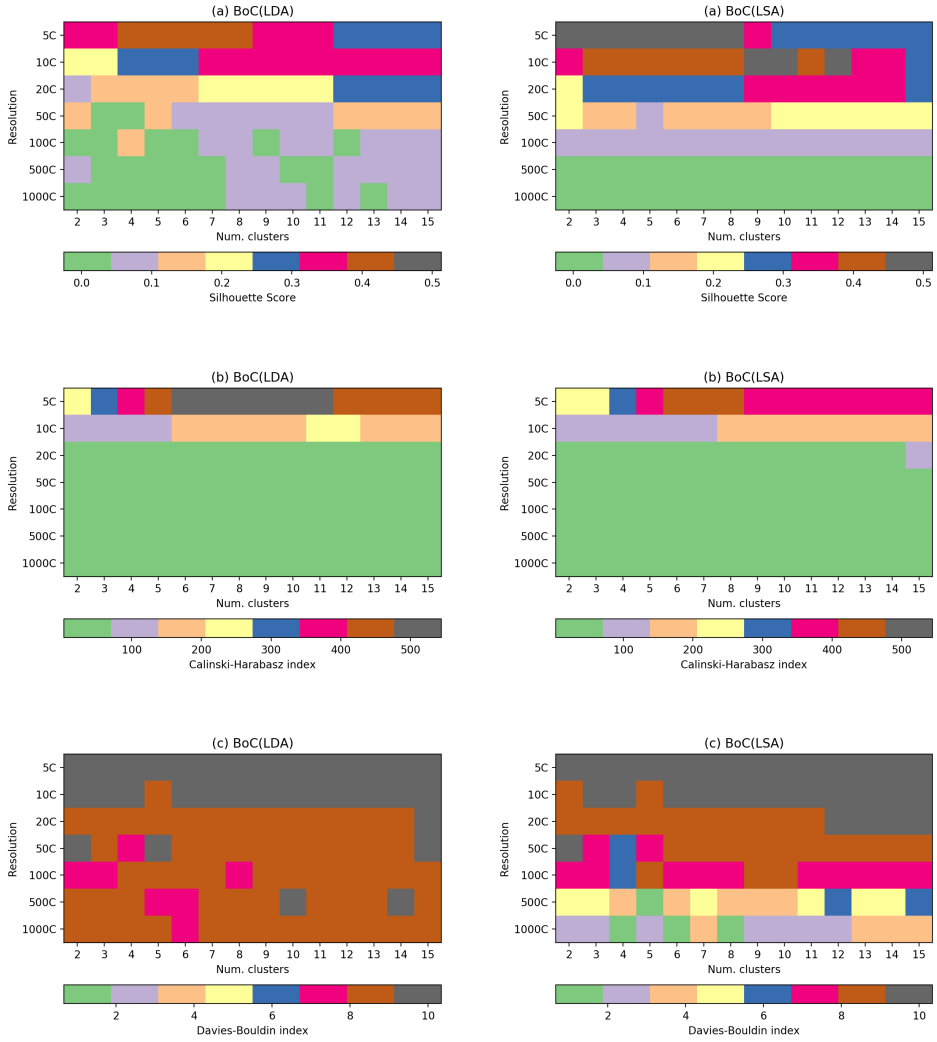


Figure 3: Heatmaps showing the impact of the resolution parameter (p) in the clustering task. First row depicts results in terms of the *s* score, second row shows the CH index, and third row represents the DB index. Graphs in the same column were generated using the same approach for inferring word representations, specifically, here we are comparing LDA and LSA approaches.

across the three evaluation metrics, although is more clear for the s and CH indexes; (2) inferring word representations with LDA and LSA (Figure 3) allows us to obtain better performance across different values of k. In general, these experiments indicate that low-resolution values (5C, 10C) are preferable for obtaining the best clustering performance in the n-tv dataset.

Additionally, we evaluated our proposed method in three benchmark datasets, namely: Reuters 8 (Cardoso-Cachopo, 2007), AG's News (Zhang et al., 2015), and 10KGNAD (Schabus et al., 2017). Table 3 shows the obtained results in terms of the s score (SH), and clustering accuracy (ACA) values. It is important to mention that although these three datasets are labeled, we cannot compute the traditional Accuracy as in a supervised classification task because the k-means will assign an arbitrary label to every cluster it forms. However, what we can do is to compute the Average Clustering Accuracy (ACA) measure, which gives the accuracy of the clustering no matter what the actual labeling of any cluster is, as long as the members of one cluster are together. Traditionally, for obtaining the ACA value it is necessary to figure out what is the best setting that would yield me the maximum clustering accuracy. For our performed experiments, we used the sklearn linear_assignmen function, which uses the Hungarian algorithm to solve this problem.

As can be observed in Table 3, Boc(LDA) experiments were performed only for 5 and 10 concepts. We do not report results with a higher number of concepts because the LDA approach was not able to obtain more than 10 topics with high probability distributions, in other words, for greater values than 10 the employed LDA implementation generated empty topics for all the three datasets.

The first four rows represent the considered baselines. As can be noticed, the CNN approach performs well in the AGs News and 10KGNAD dataset, while for the R8 dataset, the traditional BoW obtains a competitive performance. In general, we can conclude that using the LDA approach for inferring the underlying semantics represents the best approach for inferring efficient highly-dense concepts. The BoC(LDA-5C) and BoC(LDA-5C) configurations obtain good results in terms of SH and ACA metrics in the R8 and AGs News datasets respectively.

6.2. Overall performance

From the previous analysis, we choose $p = 5$ as the best resolution value, since in two out of the three considered metrics, when the number of concepts is equal 5 we obtain better performances. Therefore, the next set of experiments was done using this as the number of concepts[9] and we compare our proposed approach against baselines described in section 5.2. Figure 4 shows the obtained results across the three considered evaluation metrics. Contrary to the previous section, here we kept the

[9] Represented as the '-5C' suffix in the experiments.

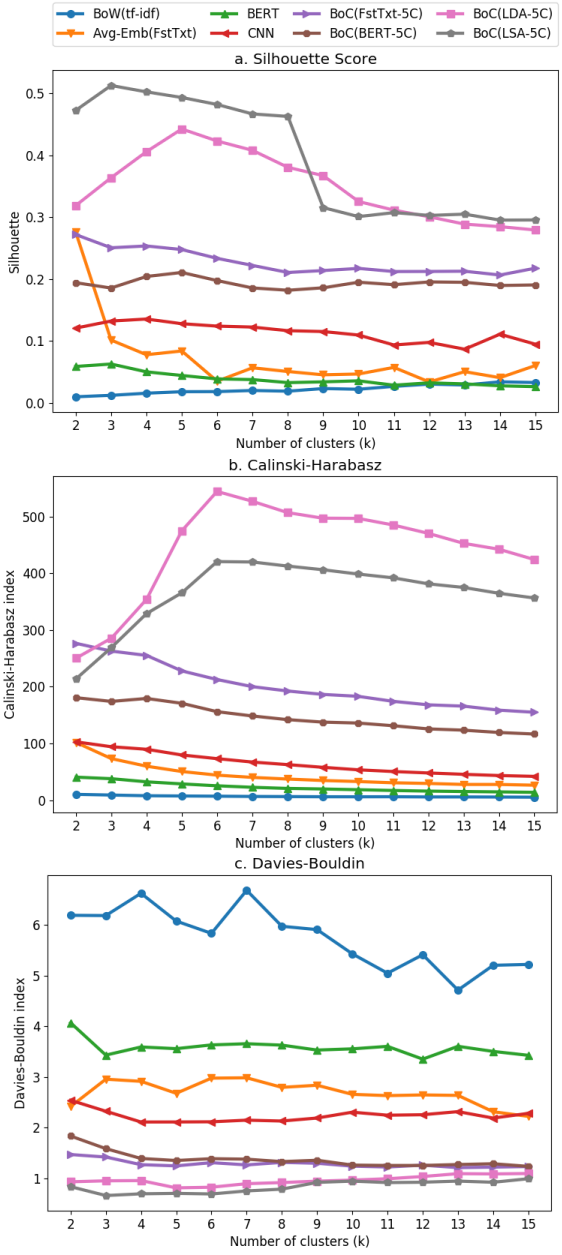


Figure 4: Clustering performance across several values of k: (a) s score, (b) CH index, and (c) DB index

Model	R8		AGs News		10KGNAD	
	SH	ACA	SH	ACA	SH	ACA
BOW	0.055	0.641	0.012	0.271	0.020	0.424
Avg-Emb(FstTxt)	0.054	0.474	0.042	0.409	0.223	0.225
BERT	0.077	0.378	0.041	0.599	0.039	0.368
CNN	0.079	0.407	0.057	0.623	0.158	**0.618**
BoC(FstTxt-5C)	0.279	0.312	0.300	0.361	0.221	0.327
BoC(FstTxt-10C)	0.199	0.325	0.203	0.344	0.231	0.513
BoC(FstTxt-20C)	0.131	0.322	0.158	0.403	0.185	0.503
BoC(FstTxt-50C)	0.098	0.319	0.122	0.579	0.116	0.495
BoC(FstTxt-100C)	0.088	0.364	0.086	0.539	0.073	0.485
BoC(FstTxt-500C)	0.057	0.392	0.043	0.610	0.034	0.527
BoC(FstTxt-1000C)	0.066	0.446	0.030	0.597	0.025	0.499
BoC(LSA-5C)	0.162	0.453	0.236	0.457	0.225	0.476
BoC(LSA-10C)	0.304	0.583	0.183	0.484	0.236	0.471
BoC(LSA-20C)	0.262	0.595	0.095	0.454	0.179	0.421
BoC(LSA-50C)	0.149	0.619	0.158	0.292	0.127	0.427
BoC(LSA-100C)	0.133	0.633	0.057	0.292	0.073	0.448
BoC(LSA-500C)	0.056	0.701	0.050	0.396	0.005	0.418
BoC(LSA-1000C)	0.085	0.592	-0.010	0.459	0.027	0.453
BoC(LDA-5C)	0.349	0.504	**0.424**	**0.793**	0.341	0.455
BoC(LDA-10C)	**0.388**	**0.721**	0.237	0.617	**0.384**	0.495

Table 3: Additional experiments on three benchmark datasets. Results are reported in terms of Silhouette score (SH), and average clustering accuracy (ACA).

original configuration of the DB index, i.e., the lower the obtained score, the better the performance of the clustering approach.

Notice that traditional BoW(tf-idf) and Avg-Emb(FstTxt) techniques obtain the worst performance overall. Similarly, the BERT approach, which represents each document using the produced encoded by the last hidden layer of the pre-trained model of BERT, obtains comparable results to those from the Avg-Emb(FstTxt) technique. Although the CNNs method (Xu et al., 2015) improves the performance of the three previous baselines, its obtained results are far from reaching those obtained with the different configurations of our proposed approach.

From these experiments, it becomes clearer that the proposed approach performs better when concepts are inferred using either LDA or LSA techniques. If we concentrate on the s score only, the best performance is obtained when using BoC(LSA-5C)

at $k = 3$ ($s = 0.51$), which represents a relative improvement of 73% against the best baseline, i.e., the CNN approach. Similarly, if we observe the CH index, the best result is obtained with BoC(LDA-5C) at $k = 6$ ($CH = 544.19$), which represents a relative improvement of 81.1% against the best result of the CNN approach. And finally, in terms of the DB index, the best performance is obtained with BoC(LSA-5C) at $k = 3$ ($DB = 0.66$), which represents a relative improvement of 68% in comparison to the CNN approach. Hence, the main observations from this analysis are: (*1*) proposed approach consistently improves, across three different metrics, traditional clustering techniques as well as some more recent approaches based on deep NN; (2) LDA and LSA techniques allow inferring better word representations, improving clustering results in comparison to SOTA methods such as BERT encodings.

6.3. Manual evaluation

To judge the quality of the generated groups, we have taken a subset of 30 articles and performed a small manual annotation experiment using 6 human experts.

For this exercise, we randomly select 30 articles from the n-tv dataset. Every annotator was instructed to identify 5 different clusters, i.e., they had to organize the information into five semantically related groups. The only restriction given is that each group should have at least one document and the same document can not be assigned to more than one cluster. We choose 5 as the number of clusters to identify, as from the previous experiments (see Figure 4) we observed that with $k = 5$ as a middle point, it is possible to obtain good performance on all the considered metrics. We evaluated the annotator's agreement using the Kappa metric (Cohen, 1968). Resulting in a Kappa score of **0.49** which indicates a moderate agreement.

We performed a detailed analysis of the identified groups, and it was clear from the exercise that spotted topics were: 'technology', 'economy', 'politics', 'car industry', and 'financial education'. We observed that annotators tend to disagree on the class of the document when the categories might be related to 'economy', 'politics', and 'financial education', similarly when a document might belong to 'technology' and 'car industry'. However, using a majority vote scheme, we decided on the final class of each document, and we used these 30 documents as a test set. We evaluate our method using the BoC(LDA-5C) configuration, and we were able to obtain a **70%** accuracy in the classification process. In Figure 5 we show the clusters' visualization under this configuration.

7. Conclusions

In this paper, we proposed using highly dense representations, denominated low-resolution concepts, for clustering German broadcast media contents. The proposed approach infers the fundamental semantic elements contained in the input dataset, which are used for suggesting optimal clusters configuration. Performed experiments

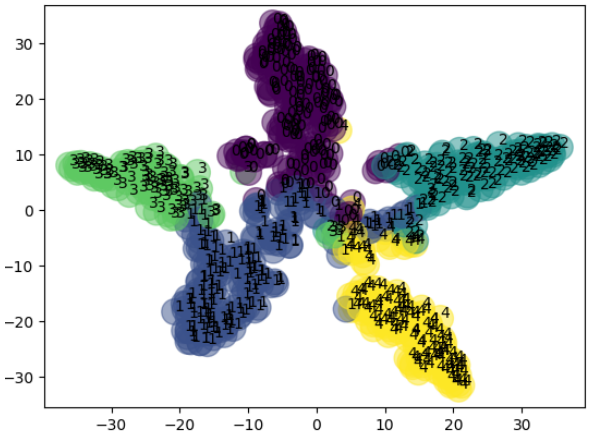


Figure 5: Formed clusters using the BoC(LDA-5C) configuration with k = 5. Found topics with the LDA approach are: *i*) chef (boss), autos (cars), deutschland (germany), zukunft (future), diesel (diesel); *ii*) euro (euro), prozent (percent), geld (money), experten (experts), deutschland (germany); *iii*) unternehmen (company), usa (USA), milliarden (billions), trump (Trump), eu (EU); *iv*) kunden (customers), google (Google), mitarbeiter (employees), online (online), facebook (facebook); and *v*) startup (sartup), deutschland (germany), daten (data), idee (idea), welt (world).

demonstrate that using small resolution values provides a better clustering performance, which is consistent across three different internal evaluation metrics, and in four different datasets. Particularly, the proposed framework is not dependent of any particular concise semantic analysis method for inferring concepts; however, when concepts are detected using the LDA and LSA approaches, the clustering performance tends to improve, obtaining relative improvements of 73%, 81%, and 68% under Silhouette, Calinski-Harabasz, and Davies-Bouldin indexes respectively. Finally, we would like to highlight one major advantage of our proposed approach, which is interpretability. As a result of the representation process, produced vectors are easy to interpret, facilitating end users understanding the found semantics and the decisions made by the system.

As future work, we plan to evaluate our proposed approach in similar datasets, i.e., very short texts, from a very narrow domain, and as the result of automatic transcription process from spontaneous speech. Is it possible to imagine, the latter represents a more challenging scenario since automatic transcription systems have many errors that might affect the performance of text-based methods.

Acknowledgments

We are very grateful to the annotators who help us manually validating the topics in the n-tv dataset: Alicia Illi, Noémi Stalder, Luca Schöb, Jasmin Staubli, Chaira Tremml, Angela Mürner. We also like to thank Daniele Zuccheri, and Jan Aeberli for providing access to the n-tv transcripts in the first place.

The work was supported by an innovation project (under an InnoSuisse grant) oriented to improve the automatic speech recognition and natural language understanding technologies for German. Title: "SM2: Extracting Semantic Meaning from Spoken Material" funding application no. 29814.1 IP-ICT and EU H2020 project "Real-time network, text, and speaker analytics for combating organized crime" (ROXANNE), grant agreement: 833635. The first author, Esaú Villatoro-Tello is supported partially by Idiap, SNI-CONACyT, CONACyT project grant CB-2015-01-258588, and UAM-C Mexico during the elaboration of this work.

Bibliography

Adhikari, Ashutosh, Achyudh Ram, Raphael Tang, and Jimmy Lin. DocBERT: BERT for Document Classification. *CoRR*, abs/1904.08398, 2019.

Blei, David M, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003.

Bojanowski, Piotr, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146, 2017. doi: 10.1162/tacl_a_00051.

Caliński, Tadeusz and Jerzy Harabasz. A dendrite method for cluster analysis. *Communications in Statistics-theory and Methods*, 3(1):1–27, 1974. doi: 10.1080/03610927408827101.

Cardoso-Cachopo, Ana. Improving Methods for Single-label Text Categorization. PdD Thesis, Instituto Superior Tecnico, Universidade Tecnica de Lisboa, 2007.

Cohen, Jacob. Weighted kappa: nominal scale agreement provision for scaled disagreement or partial credit. *Psychological bulletin*, 70(4):213, 1968. doi: 10.1037/h0026256.

Davies, David L and Donald W Bouldin. A cluster separation measure. *IEEE transactions on pattern analysis and machine intelligence*, (2):224–227, 1979. doi: 10.1109/TPAMI.1979.4766909.

De Boom, Cedric, Steven Van Canneyt, Thomas Demeester, and Bart Dhoedt. Representation learning for very short texts using weighted word embedding aggregation. *Pattern Recognition Letters*, 80:150–156, 2016. doi: 10.1016/j.patrec.2016.06.012.

Deerwester, Scott, Susan T Dumais, George W Furnas, Thomas K Landauer, and Richard Harshman. Indexing by latent semantic analysis. *Journal of the American society for information science*, 41(6):391–407, 1990.

Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019*

Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (*Long and Short Papers*), pages 4171–4186, 2019. doi: 10.18653/v1/N19-1423. URL https://www.aclweb.org/anthology/N19-1423.

Doulaty, M, O Saz, RWM Ng, and T Hain. Automatic Genre and Show Identification of Broadcast Media. In *Proceedings of the 17th Annual Conference of the International Speech Communication Association* (*Interspeech*). ISCA, 2016. doi: 10.21437/Interspeech.2016-472.

Hinton, Geoffrey E and Russ R Salakhutdinov. Replicated softmax: an undirected topic model. In *Advances in neural information processing systems*, pages 1607–1614, 2009.

Huang, Eric H, Richard Socher, Christopher D Manning, and Andrew Y Ng. Improving word representations via global context and multiple word prototypes. In *Proc. ACL*, pages 873–882, 2012.

Kim, Han Kyul, Hyunjoong Kim, and Sungzoon Cho. Bag-of-concepts: Comprehending document representation through clustering words in distributed representation. *Neurocomputing*, 266:336–352, 2017. doi: 10.1016/j.neucom.2017.05.046.

Lai, Siwei, Liheng Xu, Kang Liu, and Jun Zhao. Recurrent convolutional neural networks for text classification. In *Twenty-ninth AAAI conference on artificial intelligence*, 2015.

Le, Quoc and Tomas Mikolov. Distributed representations of sentences and documents. In *International conference on machine learning*, pages 1188–1196, 2014.

Li, Chenliang, Haoran Wang, Zhiqian Zhang, Aixin Sun, and Zongyang Ma. Topic modeling for short texts with auxiliary word embeddings. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, pages 165–174, 2016. doi: 10.1145/2911451.2911499.

Li, Zhixing, Zhongyang Xiong, Yufang Zhang, Chunyong Liu, and Kuan Li. Fast text categorization using concise semantic analysis. *Pattern Recognition Letters*, 32(3):441–448, 2011. doi: 10.1016/j.patrec.2010.11.001.

López-Monroy, Adrian Pastor, Fabio A González, Manuel Montes, Hugo Jair Escalante, and Thamar Solorio. Early text classification using multi-resolution concept representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1* (*Long Papers*), pages 1216–1225, 2018. doi: 10.18653/v1/N18-1110.

Miao, Yishu, Lei Yu, and Phil Blunsom. Neural variational inference for text processing. In *International conference on machine learning*, pages 1727–1736, 2016.

Morchid, Mohamed and Georges Linarès. A LDA-based method for automatic tagging of Youtube videos. In *2013 14th International Workshop on Image Analysis for Multimedia Interactive Services* (*WIAMIS*), pages 1–4. IEEE, 2013. doi: 10.1109/WIAMIS.2013.6616126.

Ostendorff, Malte, Peter Bourgonje, Maria Berger, Julian Moreno-Schneider, Georg Rehm, and Bela Gipp. Enriching BERT with Knowledge Graph Embeddings for Document Classification, 2019.

Rendón, Eréndira, Itzel Abundez, Alejandra Arizmendi, and Elvia M Quiroz. Internal versus external cluster validation indexes. *International Journal of computers and communications*, 5 (1):27–34, 2011.

Ribeiro-Neto, Berthier and Ricardo Baeza-Yates. Modern information retrieval. *Addison-Wesley*, 4:107–109, 1999.

Rousseeuw, Peter J. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20:53 – 65, 1987. ISSN 0377-0427. doi: 10.1016/0377-0427(87)90125-7. URL `http://www.sciencedirect.com/science/article/pii/0377042787901257`.

Schabus, Dietmar, Marcin Skowron, and Martin Trapp. One Million Posts: A Data Set of German Online Discussions. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pages 1241–1244, Tokyo, Japan, August 2017. doi: 10.1145/3077136.3080711.

Sheri, Ahmad Muqeem, Muhammad Aasim Rafique, Malik Tahir Hassan, Khurum Nazir Junejo, and Moongu Jeon. Boosting Discrimination Information Based Document Clustering Using Consensus and Classification. *IEEE Access*, 7:78954–78962, 2019. doi: 10.1109/ACCESS.2019.2923462.

Silveira, Denys, Andr'e Carvalho, Marco Cristo, and Marie-Francine Moens. Topic modeling using variational auto-encoders with Gumbel-softmax and logistic-normal mixture distributions. In *2018 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2018. doi: 10.1109/IJCNN.2018.8489778.

Staykovski, Todor, Alberto Barrón-Cedeño, Giovanni Da San Martino, and Preslav Nakov. Dense vs. Sparse Representations for News Stream Clustering. In *Text2Story@ ECIR*, pages 47–52, 2019.

Teh, Yee W, Michael I Jordan, Matthew J Beal, and David M Blei. Sharing clusters among related groups: Hierarchical Dirichlet processes. In *Advances in neural information processing systems*, pages 1385–1392, 2005.

Wang, Rui, Xuemeng Hu, Deyu Zhou, Yulan He, Yuxuan Xiong, Chenchen Ye, and Haiyang Xu. Neural Topic Modeling with Bidirectional Adversarial Training. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 340–350, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.32. URL `https://www.aclweb.org/anthology/2020.acl-main.32`.

Xu, Jiaming, Peng Wang, Guanhua Tian, Bo Xu, Jun Zhao, Fangyuan Wang, and Hongwei Hao. Short Text Clustering via Convolutional Neural Networks. In *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*, pages 62–69, 2015. doi: 10.3115/v1/W15-1509.

Zhang, Xiang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. In *Advances in neural information processing systems*, pages 649–657, 2015.

Address for correspondence:
Shantipriya Parida
`shantipriya.parida@idiap.ch`
Idiap Research Institute
Rue Marconi 19, 1920 Martigny
Switzerland.

PBML

The Prague Bulletin of Mathematical Linguistics
NUMBER 115 OCTOBER 2020 51–82

Every Layer Counts: Multi-Layer Multi-Head Attention for Neural Machine Translation

Isaac Kojo Essel Ampomah,[a] Sally McClean,[a] Lin Zhiwei,[b] Glenn Hawe[a]

[a] School of Computing, Ulster University, Belfast, UK
[b] Mathematical Science Research Center, LG006 Lanyon Building, Queen's University, Belfast, UK

Abstract

The neural framework employed for the task of neural machine translation (NMT) usually consists of a stack of multiple encoding and decoding layers. However, only the source feature representation from the top-level encoder layer is leveraged by the decoder subnetwork during the generation of target sequence. These models do not fully exploit the useful source representations learned by the lower-level encoder layers. Furthermore, there is no guarantee that the top-level encoder layer encodes all the necessary source information required by the decoder for the target generation. Inspired by recent advances in deep representation learning, this paper proposes a *Multi-Layer Multi-Head Attention* (MLMHA) module to exploit the different source representations from the multi-layer encoder subnetwork. Specifically, the decoder is allowed a more direct access to multiple encoder layers during the target generation. This technique further improves the translation performance of the model. Also, exposing multiple encoder layers enhances the flow of gradient information between the two subnetworks. Experimental results on two IWSLT language translation tasks (Spanish-English and English-Vietnamese) and WMT'14 English-German demonstrate the effectiveness of allowing the decoder access to representations from multiple encoder layers. Specifically, the MLMHA approaches explored in this paper achieve improvements up to +0.71, +0.75 and +0.49 BLEU points over the Transformer baseline model on the English-German, Spanish-English, and English-Vietnamese translation tasks respectively.

1. Introduction

Neural machine translation (NMT) architectures (Luong et al., 2015; Vaswani et al., 2017; Gehring et al., 2017) have achieved significant improvement over statistical ma-

 Corresponding author: ampomah-i@ulster.ac.uk
Cite as: Isaac Kojo Essel Ampomah, Sally McClean, Lin Zhiwei, Glenn Hawe. Every Layer Counts: Multi-Layer Multi-Head Attention for Neural Machine Translation . The Prague Bulletin of Mathematical Linguistics No. 115, 2020, pp. 51–82. doi: 10.14712/00326585.005.

chine translation techniques (Och et al., 1999; Callison-Burch et al., 2011; Koehn and Schroeder, 2007) without the need for extensive feature engineering. The backbone of these architectures is the encoder-decoder framework. The task of the encoder subnetwork is the generation of the semantic information from the source sequence. On the other hand, the decoder is charged with the target sequence generation based on the source semantic representation captured by the encoder.

Recent state-of-the-art (SOTA) NMT models (Vaswani et al., 2017; Gehring et al., 2017) implement each of the encoder and decoder subnetworks as a stack of multiple layers. The propagation of information between the two subnetworks becomes difficult as the number of layers increases. To minimize this problem, recent models (Vaswani et al., 2017; Gehring et al., 2017) employ shortcut connections such as residual units (He et al., 2016) between the layers to enhance the flow of information across the multiple layers. Furthermore, recent works (Raganato et al., 2018; Belinkov et al., 2017) also have revealed that each encoding layer extracts different levels of abstraction of the source representation. For example, Belinkov et al. (2017) evaluated representations extracted from different encoder layers on tasks such as part-of-speech tagging (POS) and semantic tagging. They argue that the lower-level encoder layers focus more on learning word-level information/properties whilst the higher-level layers encode more semantic information. All these representations can be exploited to further improve the task of sequence to sequence (seq2seq) generation. However, current NMT models generate the target sequence based on representation from only the top-level encoding layer. These models fail to fully explore the multiple useful source representations generated in the lower-level encoder layers during the target generation. A problem with this approach is that there is little to no guarantee that the necessary source information required by the decoder subnetwork is encoded in the top-level encoder layer (Wang et al., 2018; Dou et al., 2018).

Research works from the field of computer vision (Yu et al., 2018; Huang et al., 2017) have proven the benefits and the performance impact of exploiting representations from multiple top-level and lower-level layers. Inspired by this, several feature aggregation techniques have been proposed to improve the performance of NMT models (Dou et al., 2018; Wang et al., 2018; Bapna et al., 2018). These aggregation approaches focus on generating a single source representation as a combination of all representations from the multiple encoder layers. Even though these techniques provide a simple way to exploiting the multiple source representations, this work argues that allowing the decoder more direct access to the encoding layers can further improve the flow of gradient information and enhance the overall performance of the model. This paper is motivated by the findings in our previous work (Ampomah et al., 2019).

In our previous work, the performance of an RNN based seq2seq model was improved by performing the neural attention computations jointly across source representations from all encoding layers. The encoder employed comprised of multiple Bidirectional LSTM (BiLSTM) layers whilst the decoder consisted of a single LSTM

layer. Allowing the decoder more direct access to the stack of encoder layers significantly improved the performance of the model on the task of paraphrase generation. This work aims at enhancing the translation performance of a current SOTA model, namely the Transformer architecture (Vaswani et al., 2017), on the more challenging task of translating sentences from one language to another. Unlike (Ampomah et al., 2019), both the encoder and decoder subnetworks of the Transformer model employed in this work consist of multiple layers. Each of the decoding layers employs an encoder-decoder multi-head attention (MHA) sublayer to learn the source-target context information based on the source representation from the top-level encoder layer. To generate the contextual information based on the n source representations from the multiple encoder layers, the standard encoder-decoder multi-head attention sublayer is replaced with a MLMHA sublayer. The n source representations are aggregated by a *Source Feature Collector* module based on the outputs from the top-n encoding layers. The MLMHA module allows each decoding layer to interact with different levels of abstraction of the source sequence to further improve the translation quality. This also enhances the propagation of gradient information between the encoder-decoder subnetworks as each encoder layer receives error signals from all the decoding layers. Experimental results on two IWSLT language translation tasks (Spanish-English and English-Vietnamese) and WMT'14 English-German translation demonstrate the effectiveness of allowing each decoding layer direct access to representations from multiple encoder layers. The contributions of this work are:

- proposing the *Multi-Layer Multi-Head Attention* module which allows the decoding layers to exploit source representations captured by multiple encoding layers.
- demonstrating consistent improvement over models exploiting only the source representation from the top-level encoder layer.
- providing analysis on the encoder to understand the impact of exposing all encoder layers to the decoder subnetwork.
- providing analysis on the impact of varying the number of encoder layers outputs (n) that are considered by the MLMHA module within the decoding layers.

The remainder of the paper is organized as follows: Section 2 briefly reviews the related works and Section 3 provides a background to neural machine translation. The Multi-Layer Multi-Head Attention approaches are presented in Section 4. The experiments conducted are presented in Section 5, and the results are compared and discussed in Section 6. Also, Section 7 presents a detailed analysis performed to investigate the impact of exploiting multiple source representations from the encoder subnetwork via the MLMHA unit. Finally, the conclusion is presented in Section 8.

2. Related works

The proposed MLMHA framework is motivated by research and advances in deep representation learning. Effective propagation of gradient information across the multiple layers of a neural network can significantly improve its performance at learning a given task. To achieve this, several techniques including residual connections (He et al., 2016), highway network connections (Srivastava et al., 2015) and dense connections (Huang et al., 2017) have been extensively explored in areas such as computer vision and NLP. These approaches improve the propagation of features and error information across the multiple layers of the neural network via direct information paths between the layers. The simplicity and effectiveness of these skip-connection techniques allow for easy integration and have become the standard for SOTA models for learning problems employing neural networks. With respect to machine translation, models such as the self-attention based Transformer model (Vaswani et al., 2017), CNN based ConvS2S (Gehring et al., 2017) and LSTM/GRU based model (Wu et al., 2016) achieved SOTA performance by employing residual connections between the layers. As noted by Irie et al. (2019) and Vaswani et al. (2017), the performance of the Transformer model significantly degrades when trained without residual connections between the multiple sublayers. Across these models, source representations from the lower-level encoding layers are not considered during the target generation as only the top-level encoder layer's output is passed to the decoding subnetwork.

Making use of source representations from multiple encoding layers has been shown to improve the generalization performance of deep NMT models. To learn better source representation, (Wang et al., 2018) presents three information fusion techniques to combine representations from multiple encoding layers via a single information fusion layer. Similarly, (Dou et al., 2018) explored different representation aggregation approaches to combine source features generated from different encoder layers. To ensure that all layers capture diverse source information, they further proposed to train the neural model with a diversity promoting auxiliary learning objective. The static layer aggregation approaches from (Dou et al., 2018; Wang et al., 2018) (such as the linear feature combination method) as argued by Dou et al. (2019) sometimes ignore useful contextual information that can improve performance. In response, they propose dynamic layer aggregation with routing-by-agreement mechanisms where each decoding layer receives a different aggregation of source representations from each of the encoding layers. Similarly, Bapna et al. (2018) proposed *Transparent Attention Mechanism* where different joint source representation is generated for each decoding layer. Specifically, for a model with N encoding and M decoding layers, M different joint source representations are generated (one for each decoding layer) from the weighted combination of outputs from all the encoder layers including the word embedding layer. Via the *Transparent Attention Mechanism*, Bapna et al. (2018) were able to train (2-3x) deeper NMT models. The performance gain is

attributed to the *Transparent Attention Mechanism* easing the optimization of deeper models.

A common theme among these works is the generation of a single source feature representation as an aggregation of representations from different encoder layers. The decoder layers perform the source-target attention computations based on the aggregated joint source representation. These layer aggregation approaches provide a simplistic mechanism to enhance the source-target attention mechanism whilst improving the flow of gradient information from the decoding subnetwork to the encoding layers. In contrast, this work hypothesizes that providing the decoding network more direct access to representations from each encoding layer can further improve the performance of the model and further enhance gradient flow to each encoder layer. Specifically, this work proposes to perform the neural attention computations directly across outputs from different encoding layers via a *Multi-Layer Multi-Head Attention* module.

3. Background

The goal of a seq2seq generation model is to generate the target sequence $y = (y_1, \cdots, y_N)$ of length N given a source sequence $x = (x_1, \cdots, x_M)$ of length *M*, where x_i is the i^{th} source token and y_t is the t^{th} target word. The parameters of the model are learned by maximizing the likelihood function:

$$P(y \mid x; \theta) = \prod_{t=1}^{N} P(y_t \mid y_{<t}, x; \theta) \tag{1}$$

where $y_{<t} = y_1, \cdots, y_{t-1}$ is the generated target sub-sequence. Typically, seq2seq models employ an *encoder-decoder* architecture to model $P(y \mid x; \theta)$. The encoder generates the source semantic representation h^e from a given sentence *x*. Specifically, for each source token x_i, a distributed representation vector $e_i \in \mathbb{R}^d$, where d is the dimension of the vector, is generated by the word embedding layer. Based on the source embedding vectors $E_x = [e_1, e_2, \cdots, e_M]$, the encoder generates the hidden representation $h^e = [h^e_1, h^e_2, \cdots, h^e_M]$. The target sequence *y* is generated by the decoder based on the output of the encoder. During the decoding step *t*, the decoder computes the probability distribution of the target token y_t based on the output of the encoder and the partial target sequence $y_{<t} = y_1, \cdots, y_{t-1}$ as shown in Eq. (1).

The majority of earlier seq2seq architectures are RNN based models (Bahdanau et al., 2015; Cho et al., 2014; Ampomah et al., 2019), but recently architectures employing CNN (Gehring et al., 2017) and self-attention (Vaswani et al., 2017; Shaw et al., 2018) have gained significant attention.

3.1. The Transformer Model

In this work, all experiments and discussions are based on the recently proposed Transformer model (Vaswani et al., 2017). However, the explored attention mechanisms are applicable to other architectures including RNN (LSTM/GRU) based models (Bahdanau et al., 2015; Cho et al., 2014) and CNN (Gehring et al., 2017). The encoder and decoder subnetworks of the Transformer architecture employ attention mechanisms and a standard feed-forward network to model sequences of arbitrary length without the need for a recurrent unit or CNN. The attention operation employed across the different layers are based on the *multi-head attention* (MHA) (see Section 3.1.1).

The encoder subnetwork is composed of a stack of L identical layers. Each encoding layer consists of a *multi-head self-attention* sublayer and a *position-wise feed-forward sublayer* (FFN) sublayer. To ease training and improve performance, residual connection (He et al., 2016) and layer normalization layer (LayerNorm) (Ba et al., 2016) are employed around each sublayer. Formally, the output of each layer l (H_e^l) is computed as:

$$\begin{aligned} S_e^l &= \text{LayerNorm}\left(\text{MHA}(H_e^{l-1}, H_e^{l-1}, H_e^{l-1}) + H_e^{l-1}\right) \\ H_e^l &= \text{LayerNorm}\left(\text{FFN}(S_e^l) + S_e^l\right) \end{aligned} \tag{2}$$

where S_e^l is the output of the multi-head self-attention sublayer computed based on the source sentence representation of the preceding encoder layer $(l-1)$.

The decoder is also composed of a stack of L identical layers. Unlike the encoder subnetwork, each decoding layer consists of three sublayers. Similar to the encoding layer, it has *multi-head self-attention* and FFN sublayers, however, in between them is an *encoder-decoder MHA* sublayer. The *encoder-decoder MHA* sublayer is employed to perform attention computations over the output of the encoder H_e^L. Specifically, the output of each decoding layer l (H_d^l) is computed as:

$$\begin{aligned} S_d^l &= \text{LayerNorm}\left(\text{MHA}(H_d^{l-1}, H_d^{l-1}, H_d^{l-1}) + H_d^{l-1}\right), \\ E_d^l &= \text{LayerNorm}\left(\text{MHA}(S_d^l, H_e^L, H_e^L) + S_d^l\right), \\ H_d^l &= \text{LayerNorm}\left(\text{FFN}(E_d^l) + E_d^l\right) \end{aligned} \tag{3}$$

where S_d^l is the output of the multi-head self attention sublayer computed from the target representation from the preceding decoder layer $(l-1)$. E_d^l is the output of the multi-head encoder-decoder attention sublayer generated based on S_d^l and H_e^L. The top-level layer output (H_d^L) of the decoder is used by a linear transformation layer to generate the target sequence. Specifically, the linear transformation layer via softmax activation computes the output probability distribution over the target vocabulary.

3.1.1. Multi-Head Attention (MHA)

A neural attention mechanism is a crucial component in seq2seq architecture for many sequence generation problems including document summarization (Al-Sabahi et al., 2018) and NMT (He et al., 2018; Bahdanau et al., 2015) etc. The Transformer model uses the scale dot-product attention function. This takes three vectors as input, namely the queries Q, values V and keys K. It maps a given query and key-value pairs to an output which is the weighted sum of the values. The weights indicate the correlation between each query and key. This attention is shown as follows:

$$\begin{aligned} \text{Attention}(Q, K, V) &= \text{softmax}(\alpha)V \\ \alpha &= \text{score}\,(Q, K) \\ \text{score}(Q, K) &= \frac{Q \times K^{\intercal}}{\sqrt{d_k}} \end{aligned} \tag{4}$$

where $K \in \mathbb{R}^{J \times d_k}$ is the key, $V \in \mathbb{R}^{J \times d_v}$ is the value and $Q \in \mathbb{R}^{Z \times d_k}$ is the query. Z and J are the lengths of the sequences represented by Q and K respectively. d_k and d_v are the dimension of the key and value vectors respectively. The dimension of the query is also d_k to allow for the dot-product operation. The division of $Q \times K^{\intercal}$ by $\sqrt{d_k}$ is done to scale the result of the product operation hence stabilizing the computation (Vaswani et al., 2017). The overall attention weight distribution is obtained by applying the softmax$(\cdot)$ operation to the attention score $\alpha \in \mathbb{R}^{Z \times J}$.

For better performance, the Transformer architecture uses MHA which is composed of N_h(number of attention heads) scaled dot-product attention operations. Given the Q, K, and V, the multi-head attention is computed as follows:

$$\begin{aligned} \text{MHA}(Q, K, V) &= O \\ O &= HW_o \\ H &= \text{Concat}\,(head_1, head_2, \cdots, head_{N_h}) \\ head_h &= \text{Attention}(QW_h^Q, KW_h^K, VW_h^V) \end{aligned} \tag{5}$$

where QW_h^Q, KW_h^K, and VW_h^V are projections of the query, key and value vectors respectively for the h^{th} head. The projections are performed with the matrices $W_h^Q \in \mathbb{R}^{d_{model} \times d_k}$, $W_h^K \in \mathbb{R}^{d_{model} \times d_k}$ and $W_h^V \in \mathbb{R}^{d_{model} \times d_v}$. The inputs to the MHA$(\cdot)$ are $K \in \mathbb{R}^{J \times d_{model}}$, $V \in \mathbb{R}^{J \times d_{model}}$ and $Q \in \mathbb{R}^{Z \times d_{model}}$. $head_h \in \mathbb{R}^{J \times d_v}$ is the result of the scaled dot-product operation for the h^{th} head. The N_h scaled dot-product operations are combined by the concatenation function Concat$(\cdot)$ to generate $H \in \mathbb{R}^{Z \times (N_h \cdot d_v)}$. Finally, the output $O \in \mathbb{R}^{Z \times d_{model}}$ is generated from the projection of H using the weight matrix $W_o \in \mathbb{R}^{(N_h \cdot d_v) \times d_{model}}$. The multi-head attention has the same number of parameters as the vanilla (single-head) attention if

$$d_k = d_v = \frac{d_{model}}{N_h}$$

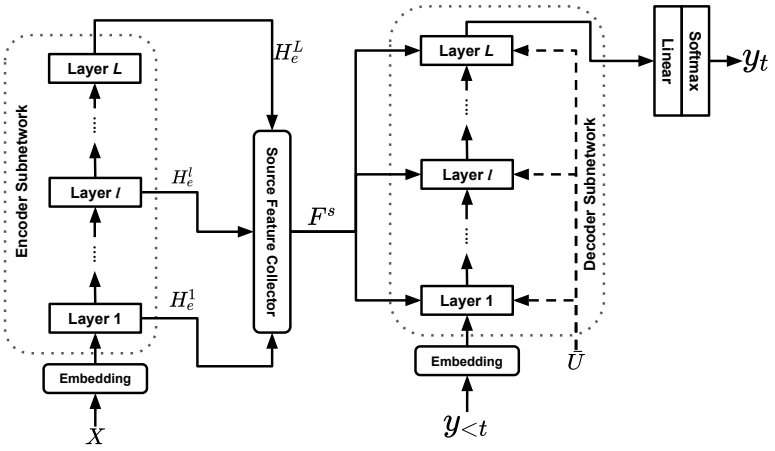


Figure 1: Illustration of the proposed approach to exploiting source representations from multiple encoding layers. X is the input sequence. y_t is the target token generated at step t and $y_{<t}$ is the generated target sub-sequence. F^s is a list of source sentence representations obtained by the *Source Feature Collector* module. The value of $\bar{U} = [\bar{U}_0, \bar{U}_1]$, (where $\bar{U}_i \in \{0, 1\}$) controls the attention computation across the source representations in F^s.

.

In a conventional encoder-decoder architecture (Vaswani et al., 2017; Gehring et al., 2017; Bahdanau et al., 2015) only the source representation from the top-level encoding layer is passed to the decoding subnetwork during the target sequence generation. As the depth of the network increases, it becomes difficult to efficiently train the model due to vanishing and exploding gradients. Furthermore, the encoder employs the entire stack of layers to learn the source semantic information. For a model with a single layer encoder subnetwork, there is a higher possibility that the top-level layer captures most of the necessary information needed to generate the target sequence. In contrast, for a deeper network, there is no guarantee that the last encoder layer's output is the best representation for the target generation due to the nature of information flow across the different time steps and the multiple layers (Wang et al., 2018; Dou et al., 2018). This work presents approaches to exploit source representations learned by multiple layers in the encoder to enhance the flow of information between the encoder and decoder subnetwork during both the forward and backward propagation.

4. Approach

The overall goal is to allow the decoder subnetwork direct access to multiple encoding layers to further enhance the translation performance of the model. To this end, each decoding layer receives a list of source sentence representations $F^s = [f^1, f^2, \cdots, f^n]$ aggregated by the *Source Feature Collector* module as shown in Fig. 1.

The *Source Feature Collector* returns a list of source representations F^s aggregated from outputs of the top n encoding layers. n is considered as a hyperparameter in this work. It is noteworthy that if $n = L$, then F^s contains out representations from all layers in a *L*-layer encoder subnetwork. The seq2seq model (Vaswani et al., 2017; Bahdanau et al., 2015; Gehring et al., 2017) using only the top-level encoder output H_e^L corresponds to setting $n = 1$. In addition to F^s, each decoder layer receives a binary vector $\bar{U} = [\bar{U}_0, \bar{U}_1]$, where $\bar{U}_i \in \{0, 1\}$. $\bar{U}$ controls how the attention computations are performed across the multiple encoder representations in F^s. Specifically, the values of $\bar{U}_0$ and $\bar{U}_1$ determine the strategy employed to generate the contextual representation base on all the source representations in F^s. In this work, four multi-layer attention strategies are explored.

Formally, the encoder-decoder multi-head attention sublayer is extended to consider the multiple source representations F^s. To this end, the encoder-decoder MHA is replaced with a MLMHA module as shown in Fig. 2. The computations across each decoder layer (see Eq. (3)) is re-formulated as follows:

$$\begin{aligned} S_d^l &= \text{LayerNorm}\left(\text{MHA}(H_d^{l-1}, H_d^{l-1}, H_d^{l-1}) + H_d^{l-1}\right), \\ E_d^l &= \text{LayerNorm}\left(\text{MLMHA}(S_d^l, F^s, \bar{U}) + S_d^l\right), \\ H_d^l &= \text{LayerNorm}\left(\text{FFN}(E_d^l) + E_d^l\right) \end{aligned} \tag{6}$$

4.1. Multi-Layer Multi-Head Attention (MLMHA)

MLMHA employs two sub-modules, namely the *Attention Aggregation Unit* and the *Context Generator*, to perform the attention computations across all representations in F^s as shown in Fig. 2. The *Attention Aggregation Unit* outputs a list of attention weights $\alpha = [\alpha^1, \alpha^2, \cdots, \alpha^n]$, where α^i is the multi-head attention weight with respect to f^i. Specifically, α^i is calculated as:

$$\begin{aligned} \alpha^i &= \text{Concat}\left(\alpha_1^i, \alpha_2^i, \cdots, \alpha_{N_h}^i\right) \\ \alpha_h^i &= \text{score}\left(QW_h^q, KW_h^k\right) \end{aligned} \tag{7}$$

where α_h^i is the attention score with respect to the attention head h and f^i. QW_h^q, and KW_h^k are, respectively, the projections of the query (S_d^l) and key (f^i) vectors for the h^{th} attention head. The projections are performed with the matrices $W_h^q \in \mathbb{R}^{d_{model} \times d_k}$, $W_h^k \in \mathbb{R}^{d_{model} \times d_k}$.

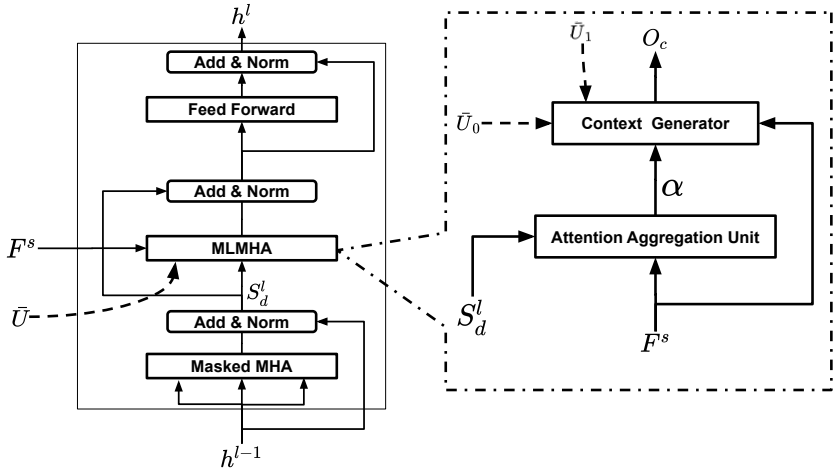


Figure 2: Illustration of a decoder layer with *Multi-Layer Multi-Head Attention* (MLMHA) sublayer to perform the attention computation across multiple features F^s received from the encoding stack. $\alpha = [\alpha^1, \alpha^2, \cdots, \alpha^n]$ is the list of attention weights (where α^i corresponds to attention weight with respect to f^i in F^s), and O_c is the joint context vector across all features in F^s.

Based on the α and F^s, the *Context Generator* computes the joint contextual representation O_c. The operation of the *Context Generator* is controlled by the values of $\bar{U}_0$, and $\bar{U}_1$. To be specific, $\bar{U}_0$ controls the generation of the context vector $c^i \in R^{Z \times d_{model}}$ with respect to f^i. Depending on the value of $\bar{U}_0$, the MLMHA module computes the c^i using either a "layer-specific-attention" weight or a "joint-attention" weight. For the case of $\bar{U}_0 = 1$, $c_h^i \in R^{Z \times d_k}$ (the context vector for the attention head h with respect to f^i) generated using the layer-specific-attention weight softmax(α_h^i) as:

$$c_h^i = \text{softmax}(\alpha_h^i) \cdot VW_h^v \tag{8}$$

where VW_h^v is the transformation of the value vector (f^i) with the projection weight $W_h^v \in \mathbb{R}^{d_{model} \times d_k}$. In contrast, for the case of $\bar{U}_0 = 0$, a joint-attention weight $\hat{\alpha}$ is employed to obtain c^i for each f^i. $\hat{\alpha}$ is calculated as:

$$\hat{\alpha} = \text{softmax}(\sum_{i=1}^{n} \alpha^i) \tag{9}$$

Analogous to Eq. (8), c_h^i is computed with $\hat{\alpha}$ as:

$$c_h^i = \hat{\alpha}_h \cdot VW_h^v \tag{10}$$

In summary, the c^i with respect to f^i is calculated as:

$$\begin{aligned} c^i &= \text{Concat}\left(c^i_1, c^i_2, \cdots, c^i_{N_h}\right) \\ c^i_h &= \begin{cases} \text{softmax}(\sum_{i=1}^{n} \alpha^i)_h \cdot VW^v_h & \bar{U}_0 = 0 \\ \text{softmax}(\alpha^i_h) \cdot VW^v_h & \bar{U}_0 = 1 \end{cases} \end{aligned} \tag{11}$$

As shown above, the c^i_h with respect to each f^i is generated using the *layer-specific-attention weight* ($\text{softmax}(\alpha^i_h)$) when $\bar{U}_0 = 1$. In contrast for $\bar{U}_0 = 0$, the c^i_h is computed with the joint-attention weight ($\text{softmax}(\sum_{i=1}^{n} \alpha^i)_h$).

Given the contexts $C = [c^1, c^2, \cdots, c^n]$ computed across source representations in F^s, a joint contextual O_c is generated as a combination of all vectors in C. The choice of combination function (either contexts-concatenation or contexts-summation) is determined by the $\bar{U}_1$. When $\bar{U}_1 = 0$, the O_c is generated from the concatenation of all the contextual representations (contexts-concatenation) in C. However for $\bar{U}_1 = 1$, O_c is obtained via the summation of the representations (contexts-summation) in C. The O_c is formulated as:

$$\begin{aligned} O_c &= \hat{C} W_o \\ \hat{C} &= \begin{cases} \text{Concat}\left(c^1, c^2, \cdots, c^n\right) & \bar{U}_1 = 0 \\ \sum_{i=1}^{n} c^i & \bar{U}_1 = 1 \end{cases} \end{aligned} \tag{12}$$

where $W_o \in \mathbb{R}^{d_c \times d_{model}}$ is the projection matrix for transforming the intermediate context representation $\hat{C} \in \mathbb{R}^{Z \times d_c}$ into $O_c \in \mathbb{R}^{Z \times d_{model}}$. It is noteworthy that the dimension size d_c is equal to d_{model} when contexts-summation ($\bar{U}_1 = 1$) is employed. In contrast, it is equal to $n \cdot d_{model}$ for contexts-concatenation ($\bar{U}_1 = 0$). In summary, the value of the binary vector $\bar{U} = [\bar{U}_0, \bar{U}_1]$ (where $\bar{U}_i \in \{0, 1\}$) presents four possible configurations of the *MLMHA* module in the decoder layer. For simplicity, the model *M-ij* denotes the configuration where $\bar{U}_0 = i$ and $\bar{U}_1 = j$ as summarized in Table 1. As shown, the *M-00* and *M-01* models generate the context c^i_h using the joint-attention weight whilst the layer-specific-attention weights are employed by the *M-10* and *M-11* models. The contexts-summation approach is employed by the M-01 and M-11 models to output contextual representation O_c. In contrast, for the M-00 and M-10 models, the contexts-concatenation approach is employed.

5. Experimental Setup

5.1. Datasets

The MLMHA strategies explored in this work are evaluated on the following language translation tasks: Spanish-English (briefly, Es-En), English-Vietnamese (briefly, En-Vi), and English-German (briefly, En-De).

Models	$\bar{U}_0$	$\bar{U}_1$	c_h^i	O_c
M-00	0	0	$\text{softmax}(\sum_{i=1}^{n} \alpha^i)_h \cdot VW_h^v$	$\text{Concat}\left(c^1, c^2, \cdots, c^n\right)$
M-01	0	1	$\text{softmax}(\sum_{i=1}^{n} \alpha^i)_h \cdot VW_h^v$	$\sum_{i=1}^{n} c^i$
M-10	1	0	$\text{softmax}(\alpha_h^i) \cdot VW_h^v$	$\text{Concat}\left(c^1, c^2, \cdots, c^n\right)$
M-11	1	1	$\text{softmax}(\alpha_h^i) \cdot VW_h^v$	$\sum_{i=1}^{n} c^i$

Table 1: Models based on the configurations of the *MLMHA* as determined by the values of $\bar{U}_0$ and $\bar{U}_1$. c_h^i is the context vector for the attention head h with respect to f^i and O_c is the overall context vector across the n source representations in F^s.

For the Es-En task, the dataset employed is from the IWSLT 2014 evaluation campaign[1] (Cettolo et al., 2014). The training set comprises of 183k training sentences pairs, and the tst 2014 split is used as the test set. The validation consisting of about 5593 sentence pairs is created by concatenating dev2010, tst2010, tst2011, and tst2012 splits. For the En-Vi translation task, the dataset is from the IWSLT 2015 English-Vietnamese track (Cettolo et al., 2015). The training set consists of 133k sentence pairs. The validation and test sets are from the TED tst 2012 (1553 sentences) and TED tst 2013 (1268 sentence pairs), respectively. For the En-De task, the models are trained on the widely-available WMT'14 dataset comprising of about 4.56 million sentence pairs for training. Following (Dou et al., 2018; Gehring et al., 2017), the newstest2013 and newstest2014 are used as the validation and test sets respectively.

To alleviate the Out-of-Vocabulary (OOV) problem, a shared vocabulary[2] generated via byte-pair-encoding (BPE)[3] (Sennrich et al., 2016) is employed to encode the source and target sentences. In the case of Es-En, the shared vocabulary comprises of about 34k sub-word tokens. For the En-Vi and En-De translation tasks, the shared vocabulary consists of 21k and 32k sub-word tokens respectively.

5.2. Model Setup

The experiments on the IWSLT tasks are conducted based on the small configuration of the Transformer architecture (Vaswani et al., 2017) with the word embedding dimension, hidden state size, and the number of attention heads set as 256, 256 and 4 respectively. The position-wise FFN has a filter of a dimension of 1024. The models trained on the Es-En and En-Vi tasks consists of a 4-layer encoder subnetwork and

[1] `https://wit3.fbk.eu/mt.php?release=2014-01`

[2] The original casing for the tokens in each sentence is preserved

[3] `https://github.com/rsennrich/subword-nmt`

4-layer decoder subnetwork. For experiments on the En-De, the base configuration is employed due to the size of the dataset. Specifically, the hidden size, filter size and the number of attention heads are 512, 2048, and 8 respectively. Both the encoder and decoder subnetworks have 6 layers. For experiments on each dataset, the value of the hyperparameter n for the *Source Feature Collector* is set to the number of layers present in the encoder i.e. $n = L$. That is, on the En-De, and IWSLT tasks n is set as 6 and 4 respectively.

5.3. Training and Inference

For the En-De task, the models are trained for 160k iterations with a batch size of 4960 tokens and a maximum sequence length is limited to 200 sub-word tokens. On the IWSLT tasks (En-Vi and Es-En), all models are trained with a batch size of 2048 tokens for a total of 200k iterations. Besides, the maximum sub-word token length is limited to 150 sub-word tokens. The optimizer employed to train the models in this work is the Adam optimizer (Kingma and Ba, 2014) (with $\beta_1 = 0.9, \beta_2 = 0.98, \epsilon = 10^9$). Following (So et al., 2019), *single-cosine-cycle* with warm-up is employed as the learning rate scheduling algorithm.

During inference, the target sentences are generated via beam search. For the IWSLT translation tasks, a beam size of 6 and a length penalty of 1.1 is employed. On the WMT'14 En-De task, the beam size of 4 and a length penalty of 0.6 is employed. common practice, the translation quality on the WMT'14 En-De, case-sensitive detokenized BLEU (Papineni et al., 2002) computed with mteval-v13a.pl[4] is employed as the evaluation metric. For the Es-En, case-sensitive BLEU metric with multi-bleu.pl[5] is used for the evaluations. Finally, the translation quality for the En-Vi is reported based on the case-sensitive BLEU score computed with sacreBLEU[6] (with the signature BLEU+case.mixed+numrefs.1+smooth.exp+tok.13a+version.1.4.13). The statistical significance is analyzed with paired bootstrap resampling (Koehn, 2004) using compare-mt[7] (Neubig et al., 2019) with 1000 resamples. The source code will be made available at `https://github.com/kaeflint/Multi-layerMHA`.

5.4. Baselines

The Transformer network (Vaswani et al., 2017) is employed as our main baseline models. However across the different languages under consideration, the performance of the MLMHA based models are compared to relevant NMT related works

[4] `https://github.com/moses-smt/mosesdecoder/blob/master/scripts/generic/mteval-v13a.pl`

[5] `https://github.com/moses-smt/mosesdecoder/blob/master/scripts/generic/multi-bleu.perl`

[6] `https://github.com/mjpost/sacrebleu`

[7] `https://github.com/neulab/compare-mt`

including *Layer Aggregation* based models. For these models, the source-target attention module of each decoder layer receives a joint source representation generated from a combination of the outputs from all the encoder layers. The joint source representation provides each decoding layer an in-direct access to multiple encoding layers. The layer aggregation approaches considered in this work are the *Linear Feature Combination* (Dou et al., 2018; Ampomah et al., 2019), the *Transparent Attention Mechanism* (Bapna et al., 2018) and the *Iterative Feature Combination* (Dou et al., 2018). For the *Linear Feature Combination*, a single joint source representation generated from the weighted combination of the outputs from the encoder layers is passed to all the layers within the decoding subnetwork. Each weight $W^l \in \mathrm{R}^{d_{model} \times d_{model}}$ controls the contribution of the l^{th} encoder layer. In contrast, for a model with N encoding and M decoding layers, *Transparent Attention Mechanism* defines single weight parameter $W \in \mathrm{R}^{(N+1)\times M}$ to generate M different joint source representations (one for each decoding layer) from the weighted combination of outputs from all the encoder layers including the word embedding layer. The *Iterative Feature Combination* proposed by Dou et al. (2018) generates the joint source representation by combining the outputs from the encoder layers in an iterative fashion starting from the lower-level layers. At each combination step s, an aggregation module consisting of a FFN, LayerNorm and residual connections is employed to merge the output from the previous step and the output from encoder layer s. Under the *Transparent Attention Mechanism* and our MLMHA, the decoder receives multiple source representations. For the *Transparent Attention Mechanism*, a joint source representation, generated from a weighted combination of the outputs from all the encoder layers is passed to each decoder layer. However, for the MLMHA mechanism, the outputs from multiple encoder layers are passed directly to each decoder layer without any modification.

6. Results

This section presents the performance evaluations of the MLMHA strategies proposed in this work on the three language translation tasks. For each language pair, the performance obtained for our MLMHA based models is compared to results from existing NMT models. The results on the WMT'14 En-De task are summarized in Table 2. For the IWSLT tasks, Table 3 and Table 4 presents the results on the Es-En and En-Vi tasks respectively. In each table, the value in parentheses represents the translation performance gain over the Transformer baseline model reimplemented in this work. On each translation task, the results obtained for each configuration of the MLMHA shows the impact of the choice of the values of $\bar{\mathrm{U}}_0$ and $\bar{\mathrm{U}}_1$.

As shown in Table 2, only the *Iterative Feature Combination* produced a statistically significant gain over the Transformer baseline among the layer aggregation approaches. The Transparent Attention produced marginal gain ($+0.13$ BLEU) whilst with the Linear Feature Combination, the performance reduced by -0.13 BLEU. For the Transformer models trained with MLMHA, the two contexts-concatenation based

Model	#Params (M)	Train	BLEU
Transformer	61.2	3.65	28.37
With Layer Aggregation			
Transformer + Linear Feature Combination	62.8	3.55	28.24 (−0.13)
Transformer + Iterative Feature Combination	77.0	3.11	28.79 (+0.42)†
Transformer + Transparent Attention	61.2	3.53	28.48 (+0.13)
With MLMHA			
M-00	92.7	2.66	28.80 (+0.43)†
M-01	84.9	2.74	28.54 (+0.17)
M-10	92.7	2.59	29.08 (+0.71)‡
M-11	84.9	2.70	28.51 (+0.14)
Existing NMT Systems			
8-Layer RNN (Wu et al., 2016)	-	-	26.30
ConvSeq2Seq (Gehring et al., 2017)	-	-	26.36
Transformer-Base (Vaswani et al., 2017)	65.0	-	27.31
Transformer+EM Routing (Dou et al., 2019)	144.8	-	28.81
Transformer+Layer Aggregation (Dou et al., 2018)	121.1	-	28.78
Layer-wise Coordination (He et al., 2018)	-	-	28.33

Table 2: Evaluation of translation performance on the WMT'14 English-German (En-De). #Params and *Train* respectively denote the number of trainable model parameters and the training speed in terms of number of steps/second. "‡" and "†" indicate statistically significant difference with $\rho < 0.01$ and $\rho < 0.05$, respectively.

models (M-00 and M-10) achieved significant gains of +0.43 BLEU and +0.71 BLEU. In contrast, the performance of contexts-summation based models (M-01 and M-11) are statistically insignificant. Table 3 summarizes the performance gains of the *M-ij* models on the IWSLT Spanish-English task. As shown, both the layer aggregation based (except the Layer Feature Combination) and our MLMHA models significantly improve the performance of the Transformer model. Compared to the layer aggregation models, our MLMHA models produced a higher gain in the translation performance. On this dataset, the overall best performance was achieved by the M-00. On the En-Vi translation task, only the M-00, M-01, Iterative Feature Combination and the Transparent Attention approaches produced significant translation quality gains.

The translation results presented in Tables 2 to 4 demonstrate the potential performance gain of leveraging source representations from multiple encoding layers. However, the improvement in translation performance is shown to be dependent on the approach employed to exploit the multiple source representations. On the En-De and Es-En tasks, providing the decoder direct access to the multiple encoder layers via the MLMHA is shown to outperform (in most cases) the indirect access provided by the layer aggregation techniques. However on the En-Vi dataset, only the M-00 and

Model	BLEU
Transformer	39.80
With Layer Aggregation	
Transformer + Linear Feature Combination	39.92 (+0.12)
Transformer + Iterative Feature Combination	40.31 (+0.51)†
Transformer + Transparent Attention	40.25 (+0.45)†
With MLMHA	
M-00	40.99 (+1.19) †
M-01	40.61 (+0.81) †
M-10	40.57 (+0.77) †
M-11	40.55 (+0.75) †
Existing NMT Systems	
UEDIN (Cettolo et al., 2014)	37.29
Tied Transformer (Xia et al., 2019)	40.51
Layer-wise Coordination (He et al., 2018)	40.50

Table 3: Evaluation of translation performance on the IWSLT Spanish-English (Es-En). "†" indicates statistically significant difference with $\rho < 0.05$.

M-01 models achieved comparable performance to the the layer aggregation models. Among our proposed models, the M-11 has the overall worse performance with the only significant gain achieved on the Es-En task. In contrast, the M-00 shows a better generalization ability as it consistently achieved statistically significant gains across the different translation tasks. The translation performance can be attributed to the joint-attention weight and contexts-concatenation techniques employed by the M-00 model as shown Table 1. The joint-attention weight is collaboratively computed across the multiple encoder layers' outputs. Compared to employing the layer-specific-attention weights, generating the context representation c^i via this strategy enhances information sharing across the encoder layers, further improving the robustness of the NMT model. Unlike contexts-summation ($\bar{u}_1 = 1$), the contexts-concatenation technique preserves much of the contextual information required for the translation task (see Section 7.2). The performance gain via the MLMHA comes at a higher computational cost in terms of the number of parameters and training speed as shown in Table 2. The layer aggregation approaches have lower impact on the training speed. For example, the Linear Feature Combination and Transparent Attention techniques degrade the speed by about 0.12 steps/second. The MLMHA introduce additional trainable parameters as each decoder layer employs n different set of weights to compute the attention weights. The M-00 and M-10 models have larger number of parameters due to the contexts-concatenation strategy. This decreases the

Model	BLEU
Transformer	30.58
With Layer Aggregation	
Transformer + Linear Feature Combination	30.91 (+0.33)
Transformer + Iterative Feature Combination	31.13 (+0.55)†
Transformer + Transparent Attention	31.16 (+0.58)†
With MLMHA	
M-00	30.88 (+0.30)†
M-01	31.07 (+0.49)†
M-10	30.71 (+0.13)
M-11	30.78 (+0.17)
Existing NMT Systems	
Luong & Manning (Luong and Manning, 2015)	23.30
NPMT (Huang et al., 2018)	27.69
NPMT + LM (Huang et al., 2018)	28.07

Table 4: Evaluation performance on the IWSLT English-Vietnamese translation task. "†" indicates statistically significant difference with $\rho < 0.05$.

training speed as more effort is required to efficiently optimize the parameters of the MLMHA based models. Section 7.2 further investigates the computational complexities of the MLMHA. Overall, based on the translation performance summarized in Tables 2 to 4, this work recommends the MLMHA with contexts-concatenation strategy to combine the contextual representations generated across the outputs from the encoder layers. The joint-attention weight technique is recommended for shallow networks of fewer number of layers, however for deeper networks, we suggest using the layer-specific-attention weight to compute the contextual representation c^i with respect to each source representation in F^s.

7. Analysis

Table 5 shows sample translations from the *M-ij* models and the Transformer baseline on the En-De translation task . This section presents further analyses performed to better understand the impact of the proposed MLMHA strategies on the performance of the Transformer model. This includes analysis to understand (a) impact on the translation quality for each *M-ij* configuration with respect to the source sentence length, (b) the impact of varying the number of source representations considered (the hyperparameter n from the *Source Feature Collector* module) on the performance of the MLMHA strategies, (c) impact on the encoder self-attention with respect to each MLMHA configuration and (d) an ablation study is conducted to understand

Source	The Aachen resident suffered serious injuries and had to be taken to the hospital for treatment.
Target	Der Aachener erlitt schwere Verletzungen und musste zur Behandlung ins Krankenhaus gebracht werden.
Baseline	Der Wohnsitz Aachen erlitt schwere Verletzungen und musste in das Krankenhaus für die Behandlung gebracht werden.
	With MLMHA
M-00	Der Aachener Resident erlitt schwere Verletzungen und musste zur Behandlung ins Krankenhaus gebracht werden.
M-01	Der Aachener Wohnsitz erlitt schwere Verletzungen und musste zur Behandlung ins Krankenhaus gebracht werden.
M-10	Der Aachener Einwohner erlitt schwere Verletzungen und musste zur Behandlung ins Krankenhaus gebracht werden.
M-11	Der Wohnsitz Aachens erlitt schwere Verletzungen und musste in das Krankenhaus gebracht werden.
Source	When the fire service arrived, the flames were already bursting out of a window.
Target	Als die Feuerwehr eintraf, schlugen die Flammen bereits aus einem Fenster.
Baseline	Als der Feuerwehr eintrat, wurden die Flammen bereits aus einem Fenster begraben.
	With MLMHA
M-00	Als der Feuerdienst eintraf, platzten die Flammen bereits aus einem Fenster.
M-01	Als der Feuerdienst eintraf, brannten die Flammen bereits aus einem Fenster.
M-10	Als der Feuerwehr eintraf, platzten die Flammen bereits aus einem Fenster.
M-11	Als der Feuerdienst ankam, brannten die Flammen bereits aus einem Fenster.

Table 5: Sample translations on the En-De task from the Transformer baseline and our MLMHA based models.

the contribution of each encoder layer to the overall translation performance of each *M-ij* model. These analyses are performed on the WMT'14 En-De due to the size of the dataset as well as the number of layers employed to train the models. For simplicity, each analysis is based on the only Transformer baseline and our *M-ij* models.

7.1. Length of source sentence

Capturing efficiently the contextual information, as well as the long-distance dependencies between the tokens of the source sentence, can significantly enhance the translation quality on longer sentences (Dou et al., 2018). Following (Luong et al.,

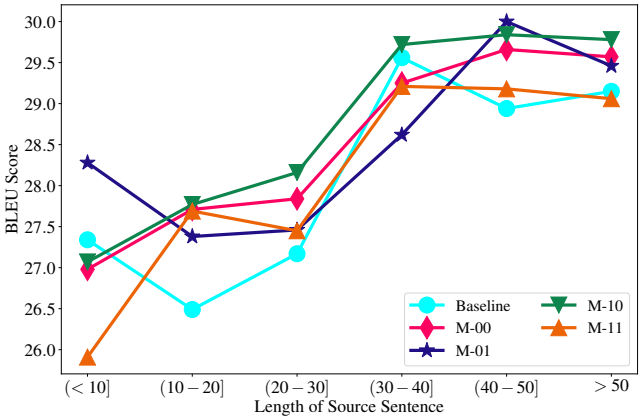


Figure 3: BLEU scores on the WMT'14 En-De test set for the Transformer baseline model, and the MLMHA models with respect to the different source sentence lengths.

2015), sentences of similar lengths (in terms of the number of source tokens) are grouped together. The choice of range for the grouping is based on the sentence lengths (the number of sub-word tokens in each source sentence) across the En-De test set. About 62% of the sentences (1,839) have sequence lengths less than 31 sub-word tokens. Therefore, the comparison presented in this section is based on the following sentence length groups: <10, 10-20, 20-30, 30-40, 40-50 and >50. For each group, the BLEU score is calculated for outputs from the models under consideration. As can be seen in the Fig. 3 , the performance of the baseline model (Transformer) generally improves with increasing input sentence lengths especially for lengths between 10 and 40 sub-word tokens. The Transformer model via the self-attention sublayers is able to model or capture the contextual information and global dependencies between the tokens irrespective of their distances or locations within the input sentence.

As shown in Fig. 3, across the sentences with lengths greater than 10, some of our models generally outperform the baseline model. This is true especially in the case of the M-10 model. It achieves the overall best translation performance for sentences longer than 20 tokens. The performance of the M-10 and M-00 models improve consistently with increasing sentence length. The M-01 achieved the best translation quality on sentences with less than 10 tokens. However, similar to the baseline, performance degrades for sentences with lengths between 10 and 20 before improving for a longer sentence. Besides, among the MLMHA models, it has the overall worse performance on sentences with lengths between 10 and 40. The M-11 model, on the other hand, performed poorly on the shorter sentences (less than 10 tokens) with the lowest BLEU score (25.91). This might explain the lower BLEU score of the contexts-summation based models (M-01 and M-11) as shown in Table 2. Overall, the perfor-

Models		#Params (M)	Train	BLEU
Baseline	B0	61.2	3.65	28.37
	B1	86.5	2.95	28.49
	B2	90.7	2.60	28.59
M-00	n=2	67.5	3.41	28.43
	n=3	73.8	3.18	28.53
	n=4	80.1	3.03	28.72
	n=5	86.4	2.77	28.66
	n=6	92.7	2.65	28.80
M-01	n=2	66.0	3.45	28.82
	n=3	70.7	3.20	28.76
	n=4	75.4	3.06	28.66
	n=5	80.1	2.93	28.46
	n=6	84.9	2.74	28.54
M-10	n=2	67.5	3.39	28.42
	n=3	73.8	3.15	28.41
	n=4	80.1	3.01	28.59
	n=5	86.4	2.76	29.12
	n=6	92.7	2.59	29.08
M-11	n=2	66.0	3.44	28.72
	n=3	70.7	3.16	28.71
	n=4	75.4	3.01	28.60
	n=5	80.1	2.83	28.62
	n=6	84.9	2.70	28.51

Table 6: Impact of n (the number of encoding layers considered by the *Source Feature Collector* module) on the performance of our MLMHA based models. B0, B1 and B2 refers to the Transformer baseline model trained with different configurations in terms of the number of layers and the filter size FFN sublayer.

mance of the *M-ij* models obtained across the different groups motivates the hypothesis that the MLMHA sublayers within the decoding subnetwork further improves the performance of the self-attention sublayers of the encoder at capturing efficiently and effectively the global dependencies between words of the input sentence. Section 7.3 explores the impact of the MLMHA on self-attention unit of each encoding layer.

7.2. Impact of the hyperparameter *n*

As shown in the Tables 2 to 4, performing the encoder-decoder multi-head attention across multiple source representations extracted from different encoding layer (in most cases) significantly improves the performance of the NMT model. This section investigates the impact of varying the value of n (i.e. using only the representations from the top n encoding layers) from 2 to 6. Specifically, each MLMHA based model is trained with different values of n. The results are summarized in Fig. 4. As seen, for all the models, there is (in most cases) a significant change in performance as the value of n increases from 1 to 6. As mentioned in Section 4, $n = 1$ correspond to the Transformer baseline model which employs output from only the top-level encoder layer.

Model Complexity

The training speed or computation speed of any given model is affected by the model size, the optimizer employed as well as any other computations that directly modify or alter the formulation of the network structure (Popel and Bojar, 2018). As shown in Section 4.1, MLMHA approach introduces new trainable parameters as each decoding layer employs n different set of weights to perform the attention computations across the multiple encoding layers. Therefore, to investigate the impact of the number of parameters on the overall training speed, we train two additional Transformer baseline models (B1 and B2) with different configurations. Specifically, the model B0 is the original Transformer from Table 2. The baseline B1 is trained with hidden size, filter size and the number of attention heads set as 512, 4098 and 8 respectively. The main difference between B0 and B2 models is that B2 employs four additional encoding and decoding layers to generate the target translations. As shown in Table 6, increasing the number of parameters generally results in a decrease in the training speed. The new parameters introduced by B1 and B2 configurations degrade the training speed by about 19.2% and 28.77% respectively.

Among the our proposed models, the M-00 and M-10 have the worst training speed compared to the M-01 and M-11 models. The number of new parameters is dependent on the strategy employed to generate the joint context O_c (see Eqs. (7) to (12)) across the multiple representations from the encoder subnetwork. When $n = 6$, the MLMHA introduces about 23.7M new parameters due the n different weights employed to perform the MHA operations across each encoder output as shown in Eq. 7 and Eq. 8. Finally for the contexts-concatenation based *M-ij* models (with $\bar{u}_1 = 0$), a further 7.8M new parameters are introduced due to the concatenation operation on the context representations $C = [c^1, c^2, \cdots, c^n]$. As shown in Table 6, for our MLMHA models, as the value of n increases, there is a corresponding reduction in the training speed from about 7.13% (when $n = 2$) to 29.04% (for $n = 6$). The configurations of the B1 and B2 models result in similar increase in the number of parameters as that of the MLMHA model (when $n = 6$). For example, the B1 model

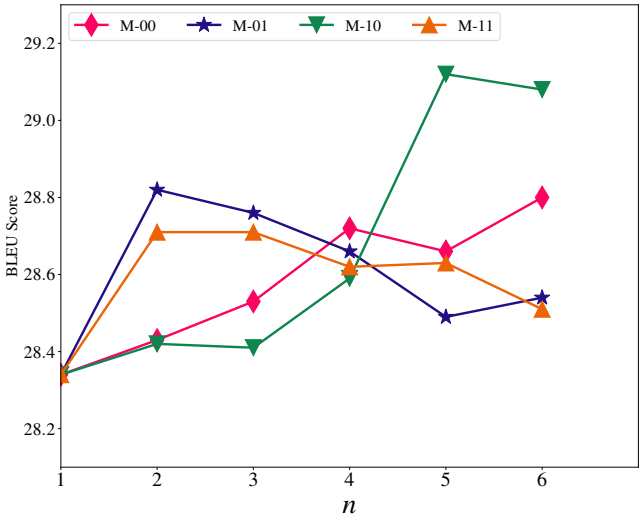


Figure 4: Variation in the translation quality across our MLMHA based models for different values of $n \leq L$.

has roughly the same number of parameters as the M-01 and M-11 models. However, the training speed of these contexts-summation models is (slightly) slower than B1. This is attributed to the additional attention computations and aggregations across the multiple encoding layers.

Translation Quality

Generally, the performance of a neural network model can be improved by either adding more layers or increasing the size of the hidden layers. As shown in Table 6, adding four encoding and decoding layers (in the case of B2) enhanced the translation quality by about $+0.2$ BLEU. For all our models, the improvement in the translation quality across the different values of n comes with an increase in the number of parameters as a result of the additional attention computations. However, unlike B1 and B2 models, we attribute the improvement in the BLEU score to the MLMHA sublayer computing a joint contextual representation O_c from the multiple source representations from the encoder. For example the M-10 (when $n = 5$) and the B1 have roughly identical number of parameters, however whilst M-10 model significantly improves the performance of B0 by $+0.75$ BLEU ($\rho < 0.01$), the B1 achieved a marginal improvement of 0.12 BLEU.

The values of n and $\bar{U}$ are shown to affect the overall translation performance of the MLMHA models. The performance of the contexts-concatenation based models (M-00 and M-10) generally improves as the number of encoder layers considered (n)

increases. These MLMHA models achieve their best performance for the values of $n > 4$ and their worst performance when $n < 4$. In contrast, the contexts-summation based models (with $\bar{U}_1 = 1$), M-01, and M-11 achieved their highest performance when $n = 2$, but the performance degrades for $n > 2$ (with the minimum BLEU score at $n = 6$ for M-11 and at $n = 5$ for the M-01 model). Specifically, the M-01 and M-11 models achieve their highest performance when only outputs from the top two encoder layers are considered. Unfortunately, these models show no statistically significant BLEU improvement over the baselines (B1 and B2) with a comparable number of parameters when $n = 6$. Among the contexts-concatenation models, only the M-10 achieved significant improvement of $+0.52$ BLEU ($\rho < 0.05$) over the B2 baseline model.

Unlike the contexts-concatenation based models, the performance of contexts-summation based models decreases as the value of n increases. This can be attributed to the fact that for the models with $\bar{U}_1 = 1$, the summation of the contextual information calculated across the source features F^s has the risk of losing some important contextual information for larger values of n. The context concatenation operation, on the other hand, preserves much of the contextual information which as shown in Table 6 improves the model's performance for larger values of n. Overall, the results obtained by the MLMHA models prove that performing the encoder-decoder attention across multiple encoder layers can further improve the performance of the NMT model. But the performance gain comes at a higher computational cost especially in the case of M-00 and M-10 models.

7.3. Impact on the Encoder's Self-attention

The performance of the encoding layers depends on the ability of the multiple heads of the self-attention unit within each layer to capture the necessary structural information. These attention heads capture structural information at varying degrees. As noted by Raganato et al. (2018) and Vig and Belinkov (2019), while some self-attention heads focus on long-distance relationships, other heads capture the shorter distance relationships between the input tokens. This allows the Transformer model to capture effectively the structural information for the given source sentence to improve the performance (Raganato et al., 2018). As stated earlier, the operations of the MLMHA module within each decoding layer affects how the source information is processed across the layer of the encoder subnetwork. Following (Vig and Belinkov, 2019), this hypothesis is tested by analyzing the attention entropy as well as the attention distance spanned by the multiple attention heads within each encoding layer's self-attention unit.

The mean distance $\bar{D}_h^l$ spanned by the attention head h with respect to the encoding layer l is computed as the weighted average distance between token pairs in all

sentences in a given corpus X. That is:

$$\bar{D}_h^l = \frac{\sum_{x \in X} \sum_{i=1}^{|x|} \sum_{y=1}^{i} w_{i,j}^h \cdot (i-j)}{\sum_{x \in X} \sum_{i=1}^{|x|} \sum_{y=1}^{i} w_{i,j}^h} \tag{13}$$

where $w_{i,j}^h$ is attention weight from the input token x_i to x_j for the attention head h. i and j denotes the locations of tokens x_i and x_j in the source sentences. Aggregating the attention distance for each head, the mean attention distance spanned $\bar{D}^l$ with respect to the encoding layer l is calculated as:

$$\bar{D}^l = \frac{1}{N_h} \cdot \sum_{h=1}^{N_h} \bar{D}_h^l \tag{14}$$

where N_h denotes the number of attention heads employed within the layer.

The mean attention distance does not offer any information on the distribution of the attention weight across the input tokens for a given attention head. The attention head with a higher mean attention distance can be concentrating on similar token sequences which might be further apart from each other (Vig and Belinkov, 2019; Ghader and Monz, 2017). To measure the concentration or the dispersion pattern of an attention head h within layer l for the input token x_i, the entropy of the attention distribution (Ghader and Monz, 2017), $E_h^l(x_i)$ for the attention head h is computed as:

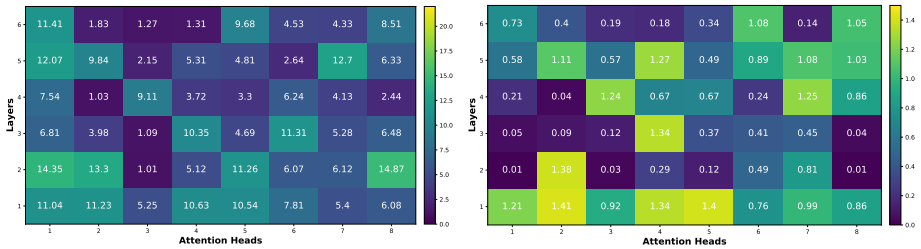
$$E_h^l(x_i) = -\sum_{j=1}^{i} w_{i,j}^h \log w_{i,j}^h \tag{15}$$

Similar to the attention distance spanned, the mean entropy of attention distribution for the encoding layer l is calculated as:

$$E^l(x_i) = \frac{1}{N_h} \sum_{h=1}^{N_h} E_h^l(x_i) \tag{16}$$

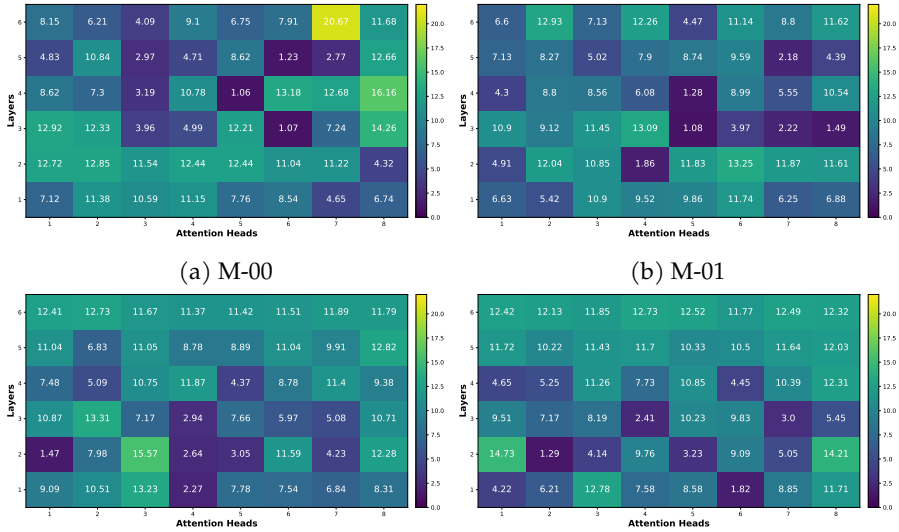
Attention heads with higher entropy are termed as having a more dispersed attention pattern while the lower the entropy, the more concentrated the attention weight distribution.

The attention distance and entropy of attention distribution analysis are performed based on the attention weights generated for 1500 randomly sampled sentences from the En-De task's test split (newstest2014). Fig. 5 and Figs. 6 and 7 show the mean attention distance span and mean entropy of attention distribution for every attention head with respect to each encoding layer for the Transformer baseline and our MLMHA models respectively. As shown, while some heads focus on the shorter-distance relationships, other heads capture the longer-distance relations among the



(a) (b)

Figure 5: Variation of of the mean attention distance span and attention distribution entropy with respect to the encoding layers and the attention heads for the Transformer baseline. (a) Mean attention distance. (b) Entropy of attention distribution.

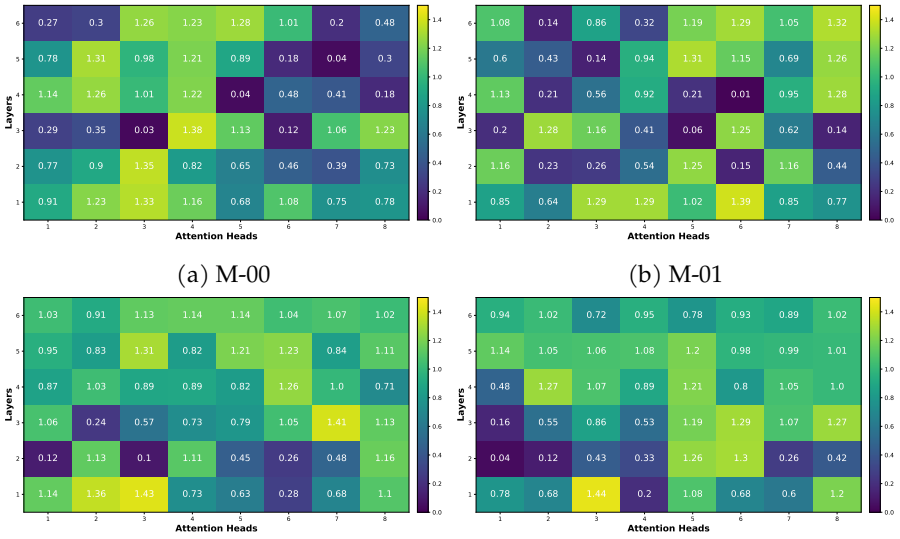


(a) M-00 (b) M-01

(c) M-10 (d) M-11

Figure 6: Variation of the mean attention distance span for the attention heads across the encoding layers with respect to the MLMHA models: (a) M-00, (b) M-01, (c) M-10, and (d) M-11.

input tokens. Similarly, the entropy of the attention distribution also varies across the layers and even for attention heads within the same layer. This is consistent with the findings of (Vig and Belinkov, 2019; Ghader and Monz, 2017). Figs. 8 and 9 show the mean average attention distance and entropy for all the self-attention heads across the layers of the encoder respectively. Each plot compares between the Transformer base-



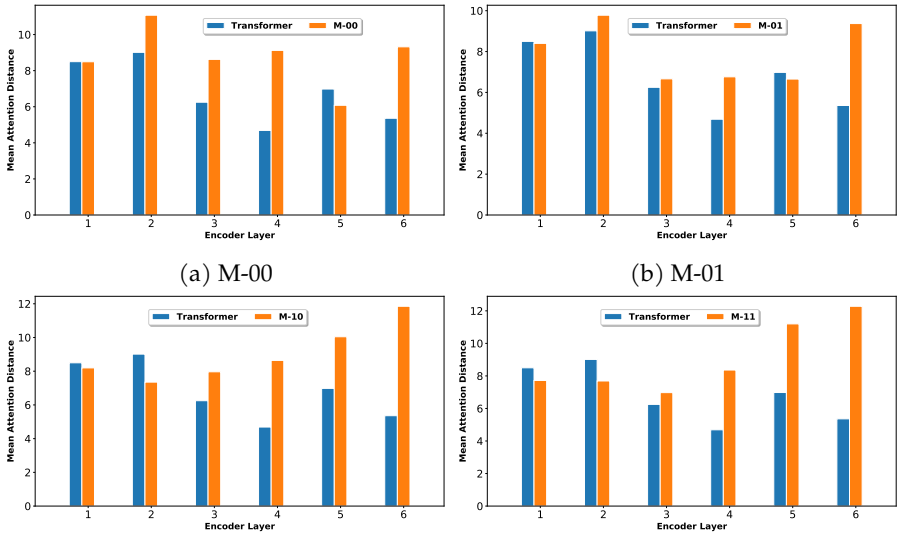
(a) M-00 (b) M-01

(c) M-10 (d) M-11

Figure 7: Variation of the entropy of attention distribution for the attention heads across the encoding layers with respect to the MLMHA models: (a) M-00, (b) M-01, (c) M-10, and (d) M-11.

line and a MLMHA model, the variations of the attention distance span and attention entropy across the encoding layers.

For the Transformer baseline, the majority of attention heads with a higher mean attention span and a more diverse attention distribution are across the first layer. But a higher mean attention distance does not always imply diverse attention distribution. In the subsequent layers, there are a number of attention heads with a higher distance span but with much more concentrated attention weights distribution. For example, layer 2 attention head 1 and head 8 have the highest mean attention spans (14.34 and 14.87 respectively) but with the lowest mean entropy scores (0.0085 and 0.0094). As noted by (Vig and Belinkov, 2019), attention heads with higher mean attention distance span concentrate their attention on words in repeated phrases at different locations within the input sentence. This could explain their lower entropy of weight distribution across the sequence of input tokens. Attention heads with diverse or concentrated weight distribution and lower attention distance span focus more on nearby tokens. Clearly, these heads with varying mean attention distance and entropy allow the Transformer to efficiently learn/capture variable structural information across its layers. This explains the superiority of the Transformer model over other seq2seq architectures such as RNN (Luong and Manning, 2015; Bahdanau et al., 2015) and CNN (Gehring et al., 2017).

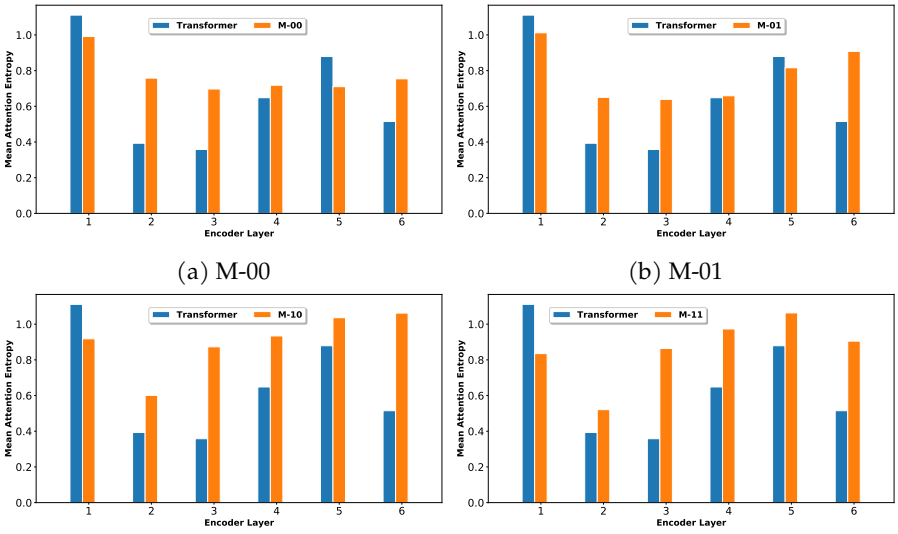


(a) M-00 (b) M-01

(c) M-10 (d) M-11

Figure 8: Variation of the average mean attention distance with respect to each encoder layer for the Transformer baseline model and our MLMHA models: (a) M-00, (b) M-01, (c) M-10, and (d) M-11. Each plot represents the average of all the attention head mean distance with respect to each encoder layer and model.

For the *M-ij* models, the impact of the different MLMHA approaches (employed by the decoder subnetwork) on the self-attention unit within the associated encoding layer is of greater interest. As shown in Figs. 6 to 9, exposing all the encoding layers to the decoding subnetwork can alter how the source information is learned across the encoder subnetwork. The change in terms of the average mean attention distance span and entropy of attention weight distribution for the multiple attention heads across the different encoder layers is dependent on the value of the $\bar{U}_0$ as evident from Figs. 8 and 9. For example, as displayed in Figs. 9a and 9b and Figs. 8a and 8b, the joint-attention weight models (M-00 and M-01) have concentrated attention heads with shorter attention distance span across the intermediate layers $3 \leq l \leq 5$. These intermediate layers are used to learn the short-range (local) contextual information within the neighborhood of the input source tokens. In contrast, the layer-specifi-atention weight models (M-10 and M-11) employs the first few layers ($l \leq 3$) to learn the short-term information whilst the upper layers model the long distance interaction between the input tokens as shown in Figs. 9c and 9d and Figs. 8c and 8d. Overall, each MLMHA strategy is shown to modify how source information is captured across the multiple attention heads and layers in the encoder as shown by attention distance and entropy of attention weight distribution. This further enhances the net-



(a) M-00 (b) M-01

(c) M-10 (d) M-11

Figure 9: Variation of the average entropy of head attention distribution across the encoding layers for the Transformer baseline model and our MLMHA models: (a) M-00, (b) M-01, (c) M-10, and (d) M-11. Each plot represents the average entropy of head attention distribution per encoder layer.

work's performance at learning the source semantic information needed to improve the translation quality.

7.4. An ablation study: Encoder Layer Dependency

The translation performance of the MLMHA models reported in Table 2 are based on exposing all the encoding layers to the decoder (i.e. $n = L$). However, it is worth understanding the contribution of each encoder layer to the overall performance of each model. To this end, the translation quality of each MLMHA model is evaluated while masking the entry in F^s corresponding to the encoder layer of interest. Here, masking an entry in F^s implies replacing the corresponding f^i with zeros. If the performance without the output of the encoder layer l (i.e. H_e^l) is significantly worse than the full model, then the H_e^l is clearly important. In contrast, H_e^l is considered redundant if the difference in translation performance is comparable.

Table 7 shows the difference in performance of our proposed models for each masked output of the encoder. As shown in most cases masking one of outputs of the encoder layers significantly degrades the translation quality. For example, without the output of the first encoder layer, the performance of both M-00 and M-01 model

	Models			
Layer	M-00	M-01	M-10	M-11
1	-15.25‡	-24.99‡	-0.83‡	-0.26
2	-0.49†	-0.19	-1.32‡	-1.13‡
3	-0.27	-0.09	-1.05‡	-0.83‡
4	0.03	0.13	-1.00‡	-1.43‡
5	-1.79‡	-0.81‡	-1.50‡	-1.34‡
6	-9.85‡	-1.49‡	-1.09‡	-0.10

Table 7: Difference in BLEU scores for each encoding layer masked (i.e. replacing the corresponding $f^i \in F^s$ with zeros) with respect to the MLMHA models when $n = L$. "‡" and "†" indicate statistically significant difference with $p < 0.01$ and $p < 0.05$, respectively. The base-BLEU scores for the M-00, M-01, M-10 and M-11 are 28.80, 28.54, 29.08 and 28.51, respectively.

decreases by -15.25 BLEU and -24.99 BLEU, respectively. Surprisingly without the output from the encoder layer 4, there is a marginal improvement (not statistically significant) in the translation quality of these models. Notably, the source representations from first and final encoding layers are shown to be redundant to the translation performance of the M-11 model, however, the outputs from these layers have statistically significant impact on the overall performance of the M-00, M-01 and M-10 models. Overall, the results in Table 7 demonstrates that for M-00, M-01 and M-11 models, the outputs from some of the encoder layers are redundant during testing and can be removed without significantly reducing the translation quality. Consistent with the observation in Section 7.2, the translation performance of the M-10 model is shown to be highly dependent on source representations from all encoder layers. Removing the output of any of these layers cause statistically significant change in performance.

8. Conclusion

In this work, the performance of the Transformer model is improved by exploiting multiple source representations captured by different encoding layers. Specifically, the decoding subnetwork is allowed direct access to the entire stack of encoding layers to extract better source-target contextual information. This technique also improves the flow of gradient information between the two subnetworks. Experimental results on IWSLT tasks (Spanish-English and English-Vietnamese) and on the WMT'14 English-German translation task show that the proposed MLMHA module can further improve the performance of the Transformer baseline. However, the analysis performed reveals that the performance gain is dependent on the values of the binary vector $\bar{U}$ and n (the number of encoding layers considered by the MLMHA module). Overall, the MLMHA with joint-attention weight ($\bar{U}_0 = 0$) showed better

generalization than with $\bar{U}_0 = 1$ across all the translation tasks under consideration. Further analysis also reveals that directly exposing the layers of the encoder subnetwork alters significantly how the global and local source contextual information is captured by the self-MHA sublayer employed within each encoder layer.

Future works include evaluating the performance of the *MLMHA* module on other NLP tasks such as document summarization and machine reading comprehension. Another interesting direction will consider investigating the potential performance gain from the combination of the MLMHA module and Layer Aggregation approaches such as the Transparent Attention (Bapna et al., 2018).

Bibliography

Al-Sabahi, Kamal, Zhang Zuping, and Mohammed Nadher. A hierarchical structured self-attentive model for extractive document summarization (HSSAS). *IEEE Access*, 6:24205–24212, 2018. doi: 10.1109/ACCESS.2018.2829199.

Ampomah, Isaac KE, Sally McClean, Zhiwei Lin, and Glenn Hawe. JASs: Joint Attention Strategies for Paraphrase Generation. In *International Conference on Applications of Natural Language to Information Systems*, pages 92–104. Springer, 2019. doi: 10.1007/978-3-030-23281-8_8.

Ba, Jimmy Lei, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv:1607.06450*, 2016.

Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *ICLR'15, arXiv:1409.0473*, 2015.

Bapna, Ankur, Mia Xu Chen, Orhan Firat, Yuan Cao, and Yonghui Wu. Training Deeper Neural Machine Translation Models with Transparent Attention. In *Proceedings of the 2018 Conference on EMNLP*, pages 3028–3033, 2018. doi: 10.18653/v1/D18-1338.

Belinkov, Yonatan, Lluís Màrquez, Hassan Sajjad, Nadir Durrani, Fahim Dalvi, and James Glass. Evaluating Layers of Representation in Neural Machine Translation on Part-of-Speech and Semantic Tagging Tasks. In *Proceedings of the 8th IJCNLP*, pages 1–10, 2017.

Callison-Burch, Chris, Philipp Koehn, Christof Monz, and Omar F Zaidan. Findings of the 2011 workshop on statistical machine translation. In *Proceedings of the 6th Workshop on SMT*, pages 22–64. ACL, 2011.

Cettolo, Mauro, Jan Niehues, Sebastian Stüker, Luisa Bentivogli, and Marcello Federico. Report on the 11th IWSLT evaluation campaign, IWSLT 2014. In *Proc. of IWSLT*, page 57, 2014.

Cettolo, Mauro, Jan Niehues, Sebastian Stüker, Luisa Bentivogli, and Marcello Federico. The IWSLT 2015 evaluation campaign. In *International Conference on Spoken Language*, page 57, 2015.

Cho, Kyunghyun, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv:1406.1078*, 2014. doi: 10.3115/v1/D14-1179.

Dou, Zi-Yi, Zhaopeng Tu, Xing Wang, Shuming Shi, and Tong Zhang. Exploiting Deep Representations for Neural Machine Translation. In *Proceedings of the 2018 Conference on EMNLP*, pages 4253–4262. ACL, 2018. doi: 10.18653/v1/D18-1457.

Dou, Zi-Yi, Zhaopeng Tu, Xing Wang, Longyue Wang, Shuming Shi, and Tong Zhang. Dynamic layer aggregation for neural machine translation with routing-by-agreement. *arXiv:1902.05770*, 2019. doi: 10.1609/aaai.v33i01.330186.

Gehring, Jonas, Michael Auli, David Grangier, Denis Yarats, and Yann N Dauphin. Convolutional sequence to sequence learning. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1243–1252. JMLR. org, 2017.

Ghader, Hamidreza and Christof Monz. What does Attention in Neural Machine Translation Pay Attention to? In *Proceedings of the 8th IJCNLP*, pages 30–39, 2017.

He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. doi: 10.1109/CVPR.2016.90.

He, Tianyu, Xu Tan, Yingce Xia, Di He, Tao Qin, Zhibo Chen, and Tie-Yan Liu. Layer-wise coordination between encoder and decoder for neural machine translation. In *Advances in Neural Information Processing Systems*, pages 7944–7954, 2018.

Huang, Gao, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on CVPR*, pages 4700–4708, 2017. doi: 10.1109/CVPR.2017.243.

Huang, Po-Sen, Chong Wang, Sitao Huang, Dengyong Zhou, and Li Deng. Towards neural phrase-based machine translation. *ICLR*, 2018.

Irie, Kazuki, Albert Zeyer, Ralf Schlüter, and Hermann Ney. Language Modeling with Deep Transformers. *Proc. Interspeech 2019*, pages 3905–3909, 2019. doi: 10.21437/Interspeech.2019-2225.

Kingma, Diederik P and Jimmy Ba. Adam: A Method for Stochastic Optimization. *arXiv:1412.6980*, 2014.

Koehn, Philipp. Statistical significance tests for machine translation evaluation. In *Proceedings of the 2004 conference on EMNLP*, pages 388–395, 2004.

Koehn, Philipp and Josh Schroeder. Experiments in domain adaptation for statistical machine translation. In *Proc. 2nd WMT*, pages 224–227, 2007. doi: 10.3115/1626355.1626388.

Luong, Minh-Thang and Christopher D Manning. Stanford neural machine translation systems for spoken language domains. In *Proceedings of the IWSLT*, pages 76–79, 2015.

Luong, Thang, Hieu Pham, and Christopher D. Manning. Effective Approaches to Attention-based Neural Machine Translation. In *Proc. EMNLP*, pages 1412–1421, Lisbon, Portugal, Sept. 2015. ACL. doi: 10.18653/v1/D15-1166.

Neubig, Graham, Zi-Yi Dou, Junjie Hu, Paul Michel, Danish Pruthi, and Xinyi Wang. compare-mt: A Tool for Holistic Comparison of Language Generation Systems. In *Proceedings of the 2019 Conference of the NAACL*, pages 35–41, 2019. doi: 10.18653/v1/N19-4007.

Och, Franz Josef, Christoph Tillmann, and Hermann Ney. Improved alignment models for statistical machine translation. In *Proc. of EMNLP and Very Large Corpora*, 1999.

Papineni, Kishore, Salim Roukos, Todd Ward, and Wei-Jing Zhu. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on ACL*, pages 311–318. ACL, 2002. doi: 10.3115/1073083.1073135.

Popel, Martin and Ondřej Bojar. Training Tips for the Transformer Model. *The Prague Bulletin of Mathematical Linguistics*, (110):43–70, 2018. doi: 10.2478/pralin-2018-0002.

Raganato, Alessandro, Jörg Tiedemann, et al. An analysis of encoder representations in transformer-based machine translation. In *Proc. of EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*. ACL, 2018. doi: 10.18653/v1/W18-5431.

Sennrich, Rico, Barry Haddow, and Alexandra Birch. Neural Machine Translation of Rare Words with Subword Units. In *Proc. ACL*, pages 1715–1725, 2016. doi: 10.18653/v1/P16-1162.

Shaw, Peter, Jakob Uszkoreit, and Ashish Vaswani. Self-Attention with Relative Position Representations. In *Proceedings of the 2018 Conference of the NAACL:HLT, Volume 2 (Short Papers)*, pages 464–468, 2018. doi: 10.18653/v1/N18-2074.

So, David, Quoc Le, and Chen Liang. The Evolved Transformer. In *International Conference on Machine Learning*, pages 5877–5886, 2019.

Srivastava, Rupesh Kumar, Klaus Greff, and Jürgen Schmidhuber. Highway networks. *arXiv preprint arXiv:1505.00387*, 2015.

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

Vig, Jesse and Yonatan Belinkov. Analyzing the Structure of Attention in a Transformer Language Model. *arXiv preprint arXiv:1906.04284*, 2019. doi: 10.18653/v1/W19-4808.

Wang, Qiang, Fuxue Li, Tong Xiao, Yanyang Li, Yinqiao Li, and Jingbo Zhu. Multi-layer representation fusion for neural machine translation. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3015–3026, 2018.

Wu, Yonghui, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv:1609.08144*, 2016.

Xia, Yingce, Tianyu He, Xu Tan, Fei Tian, Di He, and Tao Qin. Tied transformers: Neural machine translation with shared encoder and decoder. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 5466–5473, 2019. doi: 10.1609/aaai.v33i01.33015466.

Yu, Fisher, Dequan Wang, Evan Shelhamer, and Trevor Darrell. Deep layer aggregation. In *Proceedings of the IEEE conference on CVPR*, pages 2403–2412, 2018. doi: 10.1109/CVPR.2018.00255.

Address for correspondence:
Isaac Kojo Essel Ampomah
ampomah-i@ulster.ac.uk
School of Computing, Ulster University,
York Street, Belfast, County Antrim, BT15 1ED, United Kingdom

PBML

The Prague Bulletin of Mathematical Linguistics
NUMBER 115 OCTOBER 2020 83–104

The Design of Croderiv 2.0

Matea Filko, Krešimir Šojat, Vanja Štefanec

Faculty of Humanities and Social Sciences, University of Zagreb

Abstract

This paper deals with methods applied in the expansion and design of CroDeriv – the Croatian derivational lexicon. The first version of the lexicon contained only verbs that were segmented and analyzed for morphemes. The database is available online. In a further development, other parts-of-speech (adjectives, nouns) are imported into the lexicon. All imported lexemes are analyzed in terms of their morphological structure and word-formation patterns. Due to new parts-of-speech, and a new type of information, the modification of the database structure was necessary. Here, we present a restructured version of the database, adapted to include other POS, and to explicitly mark word-formation patterns among derivationally related lexemes. We focus on underlying principles for precise and refined queries based on various parameters through the online search interface.

1. Introduction

Croatian is a South Slavic language with very rich inflectional and derivational morphology. Whereas inflection is based almost exclusively on suffixation, various combinations of derivational affixes take part in word-formation. All morphological processes are characterized by frequent affixal as well as root allomorphy. Croatian inflectional morphology is extensively covered by several large lexica with paradigms and inflectional patterns used mainly in natural language processing (NLP) tasks such as lemmatization, morphosyntactic description (MSD) and part of speech (POS) tagging etc. The quantity of language resources dealing with word-formation is significantly smaller. This holds not only for Croatian but also for other languages worldwide. Moreover, derivational resources exist for a relatively limited number of languages, although the development of such resources has begun almost twenty years

Corresponding author: matea.filko@ffzg.hr
Cite as: Matea Filko, Krešimir Šojat, Vanja Štefanec. The Design of Croderiv 2.0. The Prague Bulletin of Mathematical Linguistics No. 115, 2020, pp. 83–104.
doi: 10.14712/00326585.006.

ago (CatVar (Habash and Dorr, 2003) for English; Démonette (Hathout and Namer, 2014) for French; DeriNet (Žabokrtský et al., 2016; Ševčíková and Žabokrtský, 2014) and Derivancze (Pala and Šmerk, 2015) for Czech; Word Formation Latin (Passarotti and Mambrini, 2012; Litta et al., 2016) for Latin; DerIvaTario (Talamo et al., 2016) for Italian; DErivBase (Bajestan et al., 2017; Zeller et al., 2013) for German and DErivBase.HR (Šnajder, 2014) for Croatian). These derivational resources generally focus on the annotation of word-formation processes within and across derivational families, i.e. among lexemes that share the same root. Generally, they do not provide the account of the morphological structure of words, i.e. they do not present their morphemic make-up. Procedures applied in their development range from automatic or semi-automatic to completely manual.

As mentioned, Croatian is a Slavic language with rich morphological processes both in terms of inflection and derivation. High-quality language resources dealing with the morphological structure and derivational relations of Croatian lexemes are useful for numerous NLP tasks, but they are also valuable in various theoretical work. In this paper[1], we present the expansion and redesign of the current version of the Croatian derivational lexicon – CroDeriv (Šojat et al., 2013).[2] Procedures applied in the building of its first version differ from those listed above: 1) this version of CroDeriv contained only verbs, i.e. other POS were not included[3]; 2) the focus was on a thorough analysis of the morphological structure of lexemes, whereas word-formation relations among them were not marked. In the second phase, CroDeriv has been expanded with words of other POS and the representation of derivational relations between base words and derivatives has been introduced. Consequently, online interface has been adapted to offer a wider range of possible queries.

The paper is structured as follows: in Section 2 we present the first version of CroDeriv and possible queries via online interface; in Section 3 we discuss how the analysis of verbal derivational families used so far can be applied to other POS, i.e. to adjectives and nouns, and extended in new directions. Section 4 presents the new structure of the database and new query parameters. In Section 5 concluding remarks and the outline of future work are given.

[1]This is an extended and significantly modified version of the paper "Redesign of the Croatian derivational lexicon" presented at the DeriMo 2019 Conference in Prague and published in the Proceedings as Filko et al. (2019).

[2]The search interface of the lexicon is available at `http://croderiv.ffzg.hr/`.

[3]See (Šojat et al., 2012) for the motivation to include only verbs in the first phase of the lexicon development.

2. Croatian derivational lexicon v1.0

The first version of CroDeriv contained ca 14,500 verbs[4] collected from two large Croatian corpora (Croatian National Corpus (Tadić, 2009), and Croatian web corpus hrWaC (Ljubešić and Klubička, 2014)) and free online dictionaries. All verbal lemmas, i.e. their infinitive forms, were automatically segmented into morphemes via a rule-based approach. The results were afterwards manually checked and edited. This procedure enabled the recognition of lexical morphemes / roots shared by various verbs as well as affixes used in their derivation. The recognition of mutual lexical morphemes enabled the creation of verbal derivational families, i.e. verbs with the same roots were grouped into derivational families accordingly. Morphological analysis of verbs also enabled the analysis of affix frequency and various combinations of derivational and lexical morphemes. Queries over such combinations are available online. Each lexical entry, i.e. verbal infinitive, is accompanied by additional information regarding its aspect. As in other Slavic languages, aspect is an inherent verbal category (Marković, 2012, 183); therefore, each verb was marked as perfective, imperfective, or bi-aspectual.[5] In cases of homography, lexical entries were disambiguated on the basis of aspectual properties and separated (one marked as imperfective, the other as perfective).

One of CroDeriv's distinctive features is the fact that lemmas are segmented into morphs, and morphs are linked to representative morphemes. The morphological segmentation of lemmas in CroDeriv consisted of two steps: 1) automatic segmentation via rules based on the list of various derivational affixes; 2) manual checking of the results necessary due to extensive homography and allomorphy of affixes and roots. In this process, we recognized and manually disambiguated all the homographic forms of various morphemes. Parallelly, we linked various allomorphs to single representative morphemes. The underlying principle for this line of processing is a two-layer approach consisting of a surface and a deep layer.

At the surface, the first step is the segmentation into morphs. The procedure enables that all allomorphs of a certain morpheme are identified and marked for

[4]This version is therefore referred to as *CroDeriV*.

[5]Verbal aspectual pairs are considered separate lemmas in Croatian. Moreover, Croatian words are limited to one inflectional suffix per word, and in case of verbal infinitives, this slot is filled with infinitive ending *-ti*. Thematic suffixes are also used in the formation of verbs from other POS, e.g. from adjectives or nouns (*pun* 'full' – *pun-**i**-ti* 'to fill$_{\text{IMPF}}$' – *is-pun-**i**-ti* 'to fulfill$_{\text{PF}}$'; *rad* 'work' – *rad-**i**-ti* 'to work$_{\text{IMPF}}$' – *za-rad-**i**-ti* 'to earn$_{\text{PF}}$'). Therefore, thematic suffixes, as ***-i-*** in *is-pun-**i**-ti*, are classified as derivational (Marković, 2012; Silić and Pranjković, 2005; Barić et al., 1995). However, some authors point out that the status of thematic suffixes is not clear. Thus, Manova (2015) recognizes following domains in the structure of Slavic word: (PREFIX)-BASE-(DERIVATIONAL SUFF)-(THEMATIC MARKER)-(INFLECTIONAL SUFF). As opposed to our approach, thematic suffixes are here neither derivational nor inflectional. However, we believe that every suffix is (more or less typical) member of the derivational or inflectional domain. Research on Croatian thematic markers has shown that they have more derivational than inflectional properties, thus, we consider them as members of the derivational domain.

their type. Possible types of morphemes recognized in Croatian lexemes are derivational prefixes, roots, derivational suffixes, inflectional suffixes, and interfixes for compounds. For example, the surface form of the verb *ispuniti* 'to fulfill, to fill out' would be presented as:

$$\left[is\right]_{\text{prefix}}\ \left[pun\right]_{\text{root}}\ \left[i\right]_{\text{derivational (thematic) suffix}}\ \left[ti\right]_{\text{inflectional (infinitive) suffix}}$$

whereas the compound verb *odobrovoljiti* 'to cheer up' is analyzed as follows:

$$\left[o\right]_{\text{prefix}}\ \left[dobr\right]_{\text{root}_2}\ \left[o\right]_{\text{interfix}}\ \left[volj\right]_{\text{root}_1}\ \left[i\right]_{\text{derivational suffix}}\ \left[ti\right]_{\text{inflectional (infinitive) suffix}}$$

At the deep layer, we link the prefixal allomorph *is* to its representative morph *iz*. The representative morph is the one from which other allomorphs can be established with the least number of morpho-phonological rules. This kind of analysis enables queries over roots and all derivatives within derivational families, but also over specific affixes and their combinations (prefixal, suffixal, and both) used in various derivational families.[6] However, this version of CroDeriv is limited in two ways: 1) it is restricted to only one POS, and 2) derivational relations between lexemes are not represented. In the following sections, we discuss how the database originally structured for the full analysis of Croatian verbal morphology was modified and expanded.

3. Croatian derivational lexicon v2.0

The expansion of CroDeriv is based on nominal and adjectival lemmas collected from corpora and online dictionaries of Croatian. We chose approx. 6,000 nouns and 1,000 adjectives according to their frequency indicated by the Croatian frequency dictionary (Moguš et al., 1999). We also used frequency lists generated by the corpus management system NoSketchEngine for both representative corpora (Croatian National Corpus and Croatian web corpus hrWaC).[7] Named entities were excluded from the list, since they are formed via non-productive word-formation patterns (Babić, 2002, 16). The obtained list of lemmas was used as a representative sample for further analysis and processing.

In order to incorporate lexemes of other POS and simultaneously mark word-

[6]The extensive statistics on roots, affixes and their combinations in Croatian is presented in Šojat et al. (2013).

[7]The procedure of collection and analysis of adjectives is thoroughly described in Filko and Šojat (2017). The number of approx. 6,000 nouns was obtained by merging the lists of 5.000 most frequent nouns from the above-mentioned sources. The methodology is explained in Filko (2020).

formation relations among them, the database needed to be restructured. The structure of the database remained morpheme-based,[8] i.e. we consider morphemes as basic meaningful units. Further, we assume that words have an internal structure. This *intra*-lexical structure is predictable to a certain degree, at least for certain POS. Following the two-layer approach discussed above, lemmas in Croderiv 2.0 are analyzed for morphs and morphemes. However, in this phase of development, we introduce a new type of information, i.e. the links indicating derivational relations between lexemes. As far as the database structure is concerned, this means that connections between base words and derivatives are explicitly marked and annotated. More details about the annotation scheme and underlying principles are given in the following sections. The introduction of new POS resulted in the expansion of derivational families already present in CroDeriv 1.0 and the establishment of new ones. The new ones are based on nominal and adjectival roots, previously not recorded in verbal families. The online interface for CroDeriv 2.0 enables graphical presentation of derivational relations. In other words, the online interface for CroDeriv 2.0 is designed to present graphical visualization of *inter*-lexical relations within derivational families. More details will be given below.
As mentioned, the morphological analysis follows the two-layered approach from Croderiv 1.0, and consists of two steps: 1) morph analysis at the surface layer, and 2) morpheme analysis at the deep layer (see Figure 1, the upper branch). However, the annotation of derivational relations among lexemes required an additional and different kind of analysis, i.e. the analysis of word-formation links and patterns. The distinction between morphological and word-formation analysis is exemplified in Figure 1. The results of word-formation analysis are available through the Croderiv 2.0 online interface. The new interface also provides information on 1) the type of the word-formation processes, and 2) affixal senses for the affixes detected in word-formation patterns of analyzed lemmas.[9] A detailed presentation of lexical entries in CroDeriv 2.0 is given in Section 3.3 below.

In the following subsections we describe these data more closely and focus on basic principles governing the morphological and the word-formation analysis applied in CroDeriv.

3.1. Morphological analysis

The morphological analysis of new lexical material consisted of 1) the manual segmentation of lexemes into morphs and morphemes, i.e. morph and morpheme analysis, and 2) the categorization of obtained results. The morphological structure

[8] As opposed to word-based approaches, cf. Stewart (2016, 5).

[9] The basic unit in our lexicon, following the approach in Croderiv 1.0 is lemma, i.e. infinitive form for verbs, nominative singular for nouns, nominative singular masculine for adjectives.

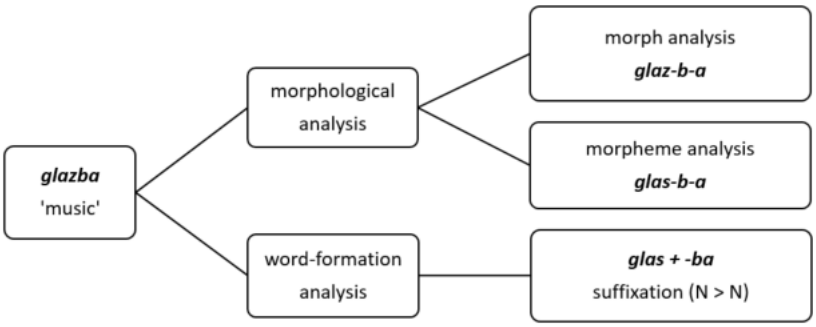


Figure 1. Morphological vs. word-formation analysis

of Croatian lexemes, regardless of their POS, consists of the following types of morphemes: prefixes, roots, interfixes, derivational and inflectional suffixes.[10] Each morpheme type can occur more than once in the morphological structure of lexemes, except inflectional suffixes. The morph and morpheme analysis is the prerequisite for the detection of both unmotivated and motivated lexemes, needed for the annotation of word-formation patterns (see Section 3.2). Generally, motivated lexemes are morphologically more complex than unmotivated, i.e. they have at least one morpheme extra in comparison to unmotivated ones. Besides, one of the aims of the manual segmentation of the representative sample is to develop a procedure for automatic segmentation in future.[11]

As opposed to verbs, usually formed via prefixation or highly-regular suffixation from other verbs (Šojat et al., 2012), nouns and adjectives are predominantly formed via suffixation. Babić (2002) lists 526 nominal and 160 adjectival suffixes out of the total of 771 suffixes used in Croatian. Although these data are useful in many aspects, the frequency of certain affixes is not provided. Frequency here refers to the number of co-occurrences of an affix and various stems as recorded in data, i.e. the number of different lexemes formed via a particular derivational affix. Preliminary research showed that a relatively small subset of suffixes compared to the numbers listed above is actually used for nominal and adjectival derivation in our representative sample.[12] As indicated, we plan to use these results for the development of a morphological parser for Croatian.

[10]Prefixes are always derivational.

[11]A procedure based on a set of rules for the detection and segmentation of single nominal suffixes was applied in Šojat et al. (2014). However, the main goal of this procedure was to detect words of the same derivational family, not to analyze their morphological structure.

[12]Filko (2020) shows that only 221 different nominal suffixes (out of 526 listed in Babić (2002)) occur in the morphological structure of 5,536 most frequent nouns in Croatian.

The morphological segmentation of new POS is based on the two-layered approach applied to verbs. At the surface layer, all possible morphs are identified and marked for their type; at the deep layer, allomorphs are connected to the single representative morph. When applied to nouns and adjectives, the analysis of the noun *učiteljica* 'female teacher' looks like this:

$$\left[uč\right]_{\text{root}} \left(\left[i\right] \left[telj\right] \left[ic\right]\right)_{\text{derivational suffixes}} \left[a\right]_{\text{inflectional suffix}}$$

whereas the adjective *izlječiv* 'curable' is segmented and processed as follows:

$$\left[iz\right]_{\text{prefix}} \left[lječ\right]_{\text{root}} \left[iv\right]_{\text{derivational suffix}} \left[\emptyset\right]_{\text{inflectional suffix}}$$

and the allomorph *lječ* is at the deep layer connected to the representative root morph *lijek*.

The analysis of morphs and morphemes is based on the following principles:[13]

- Morph analysis must be complete (no morpho-phonological residue is allowed). This means that all phonemic material of the analyzed lemma is distributed to at least one morph.
- The detection of morphs is based on commutation. This method enables the recognition of all units that 1) reoccur (e.g. *uči-**telj*** 'teacher', *vodi-**telj*** 'leader, presenter', *gleda-**telj*** 'viewer'), or 2) stand in the opposition with other units (e.g. *uč-**i**-ti* 'to learn', *hod-**a**-ti* 'to walk', *vid-**je**-ti* 'to see').
- As in other Slavic languages, numerous phonological changes occur at morpheme boundaries. Criteria for the analysis of various allomorphs, resulting from various phonological processes, are not precise. This means that in many cases it was difficult to establish straightforward links between certain parts of the phonemic material and morphs. To resolve this problem in a unified manner, the following rule was determined: if there is a fused phonemic material, allocate as much as possible of this material to the stem (see footnote 15). For example, in the word *tajništvo* 'secretariat' ← *tajnik* 'secretary' + *-stvo*, two interpretations at the surface layer are possible: *taj-n-i-štv-o* or *taj-n-iš-tvo*, depending on the allocation of the phoneme *š* to the stem or to the suffix.[14] We have decided to resolve all similar situations in favour of stems.

[13]The detailed elaboration of principles and solutions to specific problems in Croatian is given in Filko (2020).

[14]For the detailed explanation of the phonological change in this example see Marković (2013, 25, 125).

- After the surface layer morphs were detected, we determined their representative morphs at the deep layer, i.e. those to which allomorphs are connected. Hereby we follow the approach from CroDeriv 1.0: the representative morph is the one from which other allomorphs can be established with the least number of morpho-phonological rules. However, if a representative morph cannot be established via phonotactic criteria, the following frequency-based criterion is applied: the representative morph is the morph which most frequently appears in the morphological structure of various derivationally related lexemes.

The results of this analysis are reflected in the overall structure of lexical entries in CroDeriv 2.0 (see Section 3.3). As mentioned, this analysis enables the recognition of motivated lexemes and word-formation patterns. The underlying principles for the word-formation analysis are described below.

3.2. Word-formation analysis

The main goal of our word-formation analysis is to mark derivational relations among lexemes. From the theoretical point of view, for each motivated lexeme in the database we need to determine: 1) corresponding word-formation elements, and 2) a word-formation pattern. By word-formation elements in Croatian, we refer to prefixes, stems, interfixes, and suffixes. By word-formation patterns, we refer to suffixation, prefixation, simultaneous suffixation and prefixation, compounding, simultaneous compounding and suffixation, simultaneous prefixation and compounding, back-formation, and conversion / zero-derivation. Word-formation elements and patterns are presented in more detail below.

3.2.1. Word-formation elements

The first objective in the word-formation analysis is to determine word-formation elements. This step is necessary for the recognition of word-formation patterns (see Section 3.2.2). In Croatian, the following types of elements are recognized:

1. **stem**:[15] ***čaš***-*a* 'glass', where *-a* is the inflectional suffix
2. **prefix**: ***ne***-*čist* 'dirty' ← *ne-* 'non-' + *čist* 'clean'
3. **interfix**: *par*-***o***-*brod* 'steamboat' ← *par*(*a*) 'steam' + *-o-* + *brod* 'ship, boat'
4. **suffix**: *šljiv*-***ik*** 'plum yard' ← *šljiv*(*a*) 'plum' + *-ik*

[15]Following Marković (2012), we define *stem* as a segment consisting of one or more morphs to which derivational affixes are added. Stems can be equal to roots, as in *vid-jeti* 'to see' < *vid* 'sight', or they can consist of a root + one or more morphs, as in: *vidje-lica* 'psychic'< *vidjeti* 'to see'. Thus, we determine roots during the morphological analysis, and stems as a part of word-formation analysis. Derivational stems are sometimes equal to inflectional stems, e.g. *kum* 'godfather' > *kum-a* 'godmother', where *kum-* is both derivational and inflectional stem. However, in the word *čašica* 'small glass' < *čaša* 'glass', the derivational stem is *čaš-*, whereas the inflectional stem is *čašic-*. Inflectional and derivational stem are also called inflectional and derivational base.

The main problem we encountered in this analysis pertains to the status of certain suffixes. First, there are suffixes that at the same time can be interpreted as derivational as well as inflectional. For example, the suffix *-a* in the above example for the stem *čaš-* functions as a derivative suffix for the derivation of nouns, but also as an inflectional suffix for forming the nominative case, singular, femininum. Note that the main difference between this example and the thematic marker in verbs is that thematic marker is followed by an inflectional suffix (see footnote 5). Second, some suffixes that can be distinguished as different morphemes at the morphological level are added simultaneously as one derivational suffix (see the examples below). Therefore, we distinguish between (possibly complex) suffixes as word-formation elements, and (simple) suffixes in the morphological structure. The difference between them is that suffixes as word-formation elements can be morphologically complex, consisting of derivational and inflectional morphemes. Further, suffixal word-formation elements can contain more than one derivational suffix and an inflectional suffix. The motivation for this decision is twofold: 1) to resolve the status of suffixes, such as of *-a* discussed above – on the level of morphological analysis they are marked as inflectional, and 2) to indicate that groups of morphemes are simultaneously used as elements in various word-formation processes. First, we present the structure consisting of one derivational and one inflectional morpheme. The morphological structure of the Croatian noun *čistoća* 'cleanness, purity' consists of two suffixes: one derivational (*-oć-*) and one inflectional (*-a*), but there is only one word-formation suffixal element (*-oća*):

- morphological analysis (MA): *čist-oć-a*
- word-formation analysis (WFA): *čist* + *-oća* → *čistoća*

As mentioned, word-formation suffixes can consist of two (or more) derivational suffixes and one inflectional suffix (in the rightmost position). Below we list examples for adjectives, verbs and nouns, as analyzed in CroDeriv 2.0:

- *vođen* 'led'
  MA: *vođ-e-n-Ø*[16] (surface layer) *vod-je-n-Ø* (deep layer)
  WFA: *voditi* 'lead' + *-jen* → *vođen*
- *prepisivati* 'to copy'$_{\text{IMPF}}$
  MA: *pre-pis-iv-a-ti* (surface layer) *pre-pis-iv-a-ti* (deep layer)
  WFA: *prepisati* 'to copy'$_{\text{PF}}$ + *-ivati* → *prepisivati*
- *administracija* 'administration'
  MA: *administr-ac-ij-a* (surface layer) *administr-at-ij-a* (deep layer)
  WFA: *administrirati* 'to administer' + *-acija* → *administracija*

Generally, complex suffixal elements, as listed in the WFA lines above, are composed of invariant affixal combinations. Being fixed combinations, we treat them as single

[16] Zero suffix is here inflectional. Compare with genitive case: *vođ-e-n-a*.

units at this level of analysis and presentation. We intend to expand this line of research in the future. The MA lines above present the morphological analysis at the deep and the surface layer. We indicated that in many cases it is hard to determine morpheme boundaries and functions due to various phonological processes. In the following example, we demonstrate how such cases are resolved and how links between morphological and word-formation analysis are established: the noun *radništvo* 'working class' ← *radnik* 'worker' + *-stvo* consists of two word-formation elements: stem *radnik* and suffix *-stvo*. At the surface MA layer, due to morpho-phonological changes, the stem *radnik* is realized via its allomorph *radniš*, while the word-formation suffix *-stvo* is realized via its allomorph *-tvo*. At the deep layer, these allomorphs are connected to their representative forms and used for the presentation of word-formation elements. The connection of allomorphs to their representative forms at the deep layer enables the recognition of words formed via same word-formation patterns, i.e. derived via same prefixes or suffixes (e.g. *ribarstvo, radništvo* are both formed via denominal suffixation with suffix *-stvo*), and the recognition of words derived from the same stem (e.g. *radništvo, radnica, radnikov, suradnik* are formed from the stem *radnik*).

3.2.2. Word-formation patterns

Apart from word-formation elements, we also determine the type of word-formation pattern for each motivated entry in our lexicon. Word-formation patterns indicate the links between base words and various derivatives. Lexical entries provide the information on word-formation processes applied in word-formation patterns. We take into account the following word-formation processes in Croatian:

1. **suffixation:**
   - *pjev(ati)* 'to sing' + *-ač* → *pjevač* 'singer'
   - *glas* 'voice' + *-ati*[17] → *glasati* 'to vote'
   - *učitelj* 'teacher' + *-ev* → *učiteljev* 'teacher's'
2. **prefixation:**
   - *za-* + *pjev(ati)* 'to sing' → *zapjevati* 'to start singing'
   - *do-* + *predsjednik* 'president' → *dopredsjednik* 'vicepresident'
   - *pred-* + *školski* 'school'$_{\text{ADJ}}$ → *predškolski* 'preschool'$_{\text{ADJ}}$
3. **simultaneous suffixation and prefixation:**
   - *o-* + *svoj* 'one's own' + *-iti* → *osvojiti* 'to conquer, to win'
   - *bez-* + *sadržaj* 'content' + *-an* → *besadržajan* 'pointless, contentless'
4. **compounding:**
   - *vjer(a)* 'trust' + *-o-* + *dostojan* 'worthy' → *vjerodostojan* 'trustworthy'
   - *zlo* 'evil' + *upotrijebiti* 'to use' → *zloupotrijebiti* 'to misuse, to abuse'

[17] In traditional approaches, thematic suffix and infinitive ending are considered as one word-formational element consisting of two morphemes.

- *polu* 'half' + *mjesečni* 'monthly' → *polumjesečni* 'semimonthly'

5. **simultaneous compounding and suffixation:**
   - *vod*(*a*) + *-o-* + *staj*(*ati*) 'to stand' → *vodostaj* 'water level'
   - *vanjsk*(*a*) 'external' + *-o-* + *trgovin*(*a*) 'trade' + *-ski* → *vanjskotrgovinski* 'external trade'$_{\text{ADJ}}$
6. **simultaneous prefixation and compounding:**
   - *o-* + *zlo* 'evil' + *glasiti* 'to say' → *ozloglasiti* 'to discredit, to bring into disrepute'
7. **back-formation:**[18]
   - *izlaz*(*iti*) 'to exit' → *izlaz* 'exit'
8. **conversion or zero-derivation:**
   - *mlada* 'young'$_{\text{ADJ+FEM}}$ → *mlada* 'bride'$_{\text{N}}$
9. **ablaut:**
   - *plesti* = *plet* + (*Ø*) + (*ti*) 'to twine' → *plot* 'fence'.

In lexical entries, only the last step in the formation of a particular lexeme is presented. Although the verb *ispunjavati* 'to fulfill'$_{\text{IMPF}}$ is (remotely) derivationally related to the verb *puniti* 'to fill'$_{\text{IMPF}}$, their derivational connection is indirect since it is derived from the verb *ispuniti* 'to fulfill'$_{\text{PF}}$. We mark only the last derivational step in the word-formation pattern. Therefore:

ispun(*iti*) 'to fulfill'$_{\text{PF}}$ + *-javati* → *ispunjavati* 'to fulfill'$_{\text{IMPF}}$ [suffixation].

The remote derivational link is available via word-formation pattern of the verb *ispuniti* 'to fulfill'$_{\text{PF}}$:

is- + *puniti* 'to fill'$_{\text{IMPF}}$ → *ispuniti* 'to fullfil'$_{\text{PF}}$ [prefixation].

Derivational connections between motivated lexemes and their base lexemes are based on the following principle:

1. If there are simultaneous phonological and semantic relations between stems of two lexemes, two lexemes are derivationally connected (Babić, 2002, 25); e.g. *čist* 'clean' → *čist-oća* 'cleanness'.
   This principle holds in the vast majority of cases. However, in some cases stems need to be determined based on other criteria:
2. *lost* stems[19] and affixes: if a stem is synchronically not present in any other lexeme, but its suffix is clearly recognizable in the morphological structure of other

[18] Although some authors consider similar cases as examples of conversion or zero derivation (see next item), we define *conversion* as a process with no segmental or suprasegmental changes (Marković, 2012, 81). Thus, we consider cases with segmental changes as different word-formation processes. Therefore, we treat this case as back-formation as a type of subtraction.

[19] Lost stems are to be found in the so-called base-less derivatives (Gaeta and Ricca, 2003), which should synchronically be considered as simplex, since they cannot be related to any other existing base, but their suffixes are clearly recognizable from the morphological structure of the derivative in their typical senses. Lost stems are similar to the notion of unrecoverable bases (Talamo et al., 2016), and, at the word-formation level, they are similar to cranberry morphemes at the level of morphological analysis.

lexemes, this stem is taken into consideration in further processing. For example, the stem in *vrab*-ac 'sparrow' does not exist as a lexeme in Croatian, but the suffix *-ac* is normally used in the word-formation of nouns denoting male animals (e.g. *žaba* 'frog'$_{\text{FEM}}$ → *žabac* 'frog'$_{\text{MASC}}$). We refer to this type of stems as lost stems.

3. *paradigmatic* stems and affixes: if a stem cannot be synchronically associated with any existing base lexeme, but still, it occurs in at least two derivatives (Talamo et al., 2016, 84), this stem is taken into consideration in further processing. For example, the stem *dub* is recognized in the lexemes *dubok* 'deep' and *dubina* 'depth', regardless of the fact that the word *dub* does not exist. The same derivational relation is recognized in other pairs of lexemes, in which the stem functions as a separate word:
dubok 'deep' vs. *dubina* 'depth' vs. **dub*
širok 'wide' vs. *širina* 'width' vs. *šir* 'width $_{\text{expressive}}$'
visok 'high' vs. *visina* 'height' vs. *vis* 'height $_{\text{expressive}}$'.
We refer to this type of stems as paradigmatic stems.[20]
4. *possible* stems and affixes: in many derivational families, word-formation patterns cannot be established in a straightforward manner due to *missing links* between members of families. These links can be theoretically postulated as *possible* words, completely compliant to morphological structure and derivational processes in Croatian. Thus, if a base word actually does not exist, but it could be formed via regular and productive word-formation patterns, this stem is taken into consideration in further processing. Such cases are usually related to verbal participles and gerunds. In example 1 below, the past participle is attested and used for further derivation. In example 2, the past participle is not attested, i.e. it actually does not exist. However, its morphological structure is analogous to attested forms, it is marked as such and used in the database structure:[21]
1) *pjevati* 'to sing' → *pjevan* 'sung' → *pjevanje* 'singing'
2) *sjećati se* 'to remember' → **sjećan* 'remembered' → *sjećanje* 'remembrance, memory'.

In some cases, it is hard to determine the word-formation pattern due to several plausible possibilities, especially when dealing with suffixation. In these cases, we follow the criteria established in Babić (2002, 38–41):

- if one of the competing solutions increases the overall number of derivational units in Croatian, the other solution should be selected;

[20]The difference between paradigmatic and lost stems is visible in the graphical representation of derivational families - paradigmatic stems serve as the basis for the word-formation of two or more words, while only one word is derived from the lost stems.

[21]This line of processing is similar to the approach used in DeriNet 2.0 and their *fictitious lexemes*, which are defined as "lexemes that are attested neither in the corpora nor in the dictionaries but, based on structural analogies, fill a paradigm gap in the derivational family" (Vidra et al., 2019, 82).

- if one of the competing solutions can be applied to a wider range of motivated lexemes than the other, this solution should be selected.

3.2.3. Affixal senses

Morphological processing in CroDeriv enables the recognition of various combinations of affixes and roots and therefore provides an excellent basis for research. The research on the semantic impact of affixes in word-formation processes shows that derivational affixes frequently behave in a similar manner in various derivational families. In other words, derivational prefixes and suffixes similarly or even identically affect the meaning of derivatives in different derivational families. This means that the meaning structure of derivational affixes can be decomposed and its meaning components, i.e. affixal senses, can be (more or less) determined. We intend to incorporate this information into lexical entries. In our database, affixes are structured as polysemous units, which is in line with recent approaches to affixal senses (Babić (2002, 38), Lehrer (2003), Lieber (2004, 11), Lieber (2009, 41), Aronoff and Fudeman (2011, 140–141)). Taking into account other elements in word-formation patterns, one of the affixal meanings is realized in motivated lexemes. For example, the verbal prefix *nad-* can have two senses. It can express:

1. **location** (subtype: *over*), e.g. *letjeti* 'to fly' → *nadletjeti* 'to fly over'
2. **quantity** (subtype: *exceeding*), e.g. *rasti* 'to grow' → *nadrasti* 'to outgrow'.

The semantic analysis of Croatian verbal prefixes is given in Šojat et al. (2012), whereas the most frequent adjectival suffixes are discussed in Filko and Šojat (2017). A detailed semantic analysis of highly frequent nominal suffixes is presented in Filko (2020).[22] The inventory of affixal senses is based on data from Croatian grammar and reference books. As expected, affixes and their senses are treated differently in Croatian literature. Whereas some authors (e.g. Babić (2002)) list affixes alphabetically and note their possible senses, others (e.g. Silić and Pranjković (2005) and Barić et al. (1995)) list possible meanings of motivated words (e.g. diminutives, locations, instruments, male agents, female agents, animals, etc.) and indicate which affixes can be used for the creation of these meanings. In other words, they group affixes according to at least one of their meaning components. We combined the information from these sources and modified polysemous structures of affixes according to recorded lexemes in the database. For the nominal suffix *-ica* the following senses were determined (new ones may appear in future analysis):[23]

1. **agent, female**, e.g. *učitelj* 'teacher'$_{\text{MASC}}$ → *učiteljica* 'teacher'$_{\text{FEM}}$

[22] Bagasheva (2017) presents the comprehensive list of semantic categories which should be applicable for the study of affixal derivation, at least in European languages. Her set of 51 comparative semantic concepts in affixation is used as a starting point in the *Cross-linguistic research into derivational networks* project. First results of this project are presented in Körtvélyessy (2019).

[23] These are the senses recorded so far in our material. For a more extensive account, including idiosyncratic combinations, see Babić (2002, 183–189)

2. **person, both sexes**, e.g. *izbjegao* 'exiled' → *izbjeglica* 'refugee'
3. **animal, female**, e.g. *golub* 'pigeon'$_{\text{MASC}}$ → *golubica* 'pigeon'$_{\text{FEM}}$
4. **diminutive**, e.g. *pjesma* 'song' → *pjesmica* 'ditty, rhyme'
5. **thing**, e.g. *sanjar* 'dreamer'$_{\text{MASC}}$ → *sanjarica* 'dream book'
6. **drink**, e.g. *med* 'honey' → *medica* 'honey liqueur'
7. **plant**, e.g. *otrovan* 'poisonous' → *otrovnica* 'poisonous plant, mushroom (and venomous snake)'
8. **location**, e.g. *okolo* 'around' → *okolica* 'surrounding'
9. **temporal mark**, e.g. *godišnji* 'yearly' → *godišnjica* 'anniversary'
10. **disease**, e.g. *vruć* 'hot' → *vrućica* 'fever'
11. **literary type**, e.g. *slovo* 'letter' → *poslovica* 'proverb'
12. **linguistic term – type of word/sentence**, e.g. izveden 'derived'$_{\text{ADJ}}$ → izvedenica 'derivative'
13. **number of men involved**, e.g. *dvoje* 'two, of different gender' → *dvojica* 'two, of male gender'
14. **anatomical part**, e.g. *jagoda* 'strawberry' → *jagodica* 'cheekbone, fingertip'

To sum up, the new version of the database provides the information on the following word-formation properties:

- word-formation pattern: ***učiteljica*** ← *učitelj* + *ica* [suffixation]; ***izlječiv*** ← *izliječiti* + *iv* [suffixation]
- allomorph of the stem – stem: *učitelj* – *učitelj*; *izlječ* – *izliječ*
- allomorph of the affix – affix: *ica* – *ica*; *iv* – *iv*
- affix sense: agent, feminine; possibility
- POS of the stem: N; V.[24]

3.3. The structure of lexical entries

The information discussed in Sections 3.1 and 3.2 is encoded for each entry in the lexicon. The new search interface will provide the information about grammatical categories (1), morphological structure (2-3), and word-formation properties (4-8) (see the example for the lemma *poslužitelj* and others below). A link to the base word will be available through the word-formation pattern (4 - poslužiti). The list of all derivatives of the same stem will be accessible through another link attached to the stem (5 - posluži). This will enable users to follow complete derivational paths in both directions: from roots to derivatives (through the link in 4) and from various derivatives back to roots (through the link in 5). In future, we plan to provide links to online dictionaries and inflectional lexica for Croatian for additional information.

[24]This representation is in line with Babić (2002, 16), probably the most extensive and thorough book on word-formation for a Slavic language, where it is stated that derivational representation should at least show 1) word-formational units (affixes); 2) word-formational stems; 3) types of word-formation processes; 4) meanings of derived words. For the morphological analysis of these entries see Section 3.1.

The complete structure of entries of different POS is as follows:
Nouns

1. **lemma:** poslužitelj ‘server’
   - **POS:** N
   - **gender**: masculine
2. **morphological structure – surface layer:**
   $[\text{po}]_{\text{prefix}}\ [\text{služ}]_{\text{root}}\ \left(\left[\text{i}\right]\ \left[\text{telj}\right]\right)_{\text{derivational suffixes}}\ [\ ]_{\text{inflectional suffix}}$
3. **morphological structure – deep layer:**
   $[\text{po}]_{\text{prefix}}\ [\text{slug}]_{\text{root}}\ \left(\left[\text{i}\right]\ \left[\text{telj}\right]\right)_{\text{derivational suffixes}}\ [\text{Ø}]_{\text{inflectional suffix}}$
4. **word-formation pattern:** poslužiti[25] + telj
5. **stem (allomorph of the stem)**: posluži[26] (posluži)
6. **affix (allomorph of the affix)**: -telj (-telj)
7. **affix sense**: instrument
8. **word-formation process** (POS → POS): suffixation (V → N)
9. **link to the Croatian Language Portal**[27].

Verbs

1. **lemma:** potpisati ‘to sign’
   - **POS:** V
   - **aspect**: perfective
   - **reflexivity**: non-reflexive
2. **morphological structure – surface layer:**
   $[\text{pot}]_{\text{prefix}}\ [\text{pis}]_{\text{root}}\ [\text{a}]_{\text{derivational suffix}}\ [\text{ti}]_{\text{inflectional suffix}}$
3. **morphological structure – deep layer:**
   $[\text{pod}]_{\text{prefix}}\ [\text{pis}]_{\text{root}}\ [\text{a}]_{\text{derivational suffix}}\ [\text{ti}]_{\text{inflectional suffix}}$
4. **word-formation pattern:** pod + pisati
5. **stem (allomorph of the stem)**: pisati (pisati)
6. **affix (allomorph of the affix)**: pod- (pot-)
7. **affix sense**: location: under
8. **word-formation process** (POS → POS): prefixation (V → V)
9. **link to the Croatian Language Portal**.

[25]The base word is underlined and functions as a link to the entry of that word in the lexicon.

[26]The stem is underlined and functions as a link to all lemmas derived directly from this stem, e.g. *poslužilac*.

[27]Online dictionary of Croatian: `http://hjp.znanje.hr/`.

Adjectives

1. **lemma:** beskrajan ‘endless’
   - **POS:** A
   - **gender**: masculine
   - **definiteness**: indefinite
2. **morphological structure – surface layer:**
   $\left[\text{bes}\right]_{\text{prefix}}\ \left[\text{kraj}\right]_{\text{root}}\ \left[\text{an}\right]_{\text{derivational suffix}}\ \left[\ \right]_{\text{inflectional suffix}}$
3. **morphological structure – deep layer:**
   $\left[\text{bez}\right]_{\text{prefix}}\ \left[\text{kraj}\right]_{\text{root}}\ \left[\text{an}\right]_{\text{derivational suffix}}\ \left[\emptyset\right]_{\text{inflectional suffix}}$
4. **word-formation pattern:** bez + kraj + an
5. **stem (allomorph of the stem)**: kraj (kraj)
6. **affix$_1$ (allomorph of the affix$_1$)**: bez- (bes-)
   affix$_2$ (allomorph of the affix$_2$): -an (-an)
7. **affix$_1$ sense**: deprivation
   affix$_2$ sense: having the property of [meaning of the base]
8. **word-formation process** (POS → POS): simultaneous prefixation and suffixation (N → A)
9. **link to the Croatian Language Portal**.

In the following section, we focus on the redesign of the database based on the analysis of the initial set of nouns and adjectives in terms of their morphological structure and word-formation properties.

4. Redesign of the CroDeriv database

Unlike many existing derivational lexicons and databases, which mostly focus on presenting derivation as connections between lexemes and thus building derivational trees or graphs (Kyjánek et al., 2019), CroDeriv is primarily devised as a morphological resource. It means that derivational relationships are seen as a result of a specific change in the morphological structure between two lexemes, and as such recorded and presented in the database structure.

The integration of new data required a redesign of the database. The first version of CroDeriv contained only verbs and the data model was therefore built upon the generalized morphological structure of Croatian verbs. Croatian verbs, in various affixal combinations, can take up to four prefixes, three derivational suffixes, and one inflectional suffix. The lexical part contains one or two lexical stems and an optional interfix. The first data model thus provided 9 slots for affixal allomorphs, connected to their respective morphemes, and two slots for lexical stems connected to their respective forms at the deep layer. Apart from the fact that this data model could not accommodate lexemes of other POS, it suffered from other shortcomings, as well.

In this design, derivational relationships were not explicitly marked. Further, the search engine used for CroDeriv 1.0, due to simplified presentations of the generalized morphological structure, showed only stems, and not their morphological structures. These structures can be complex, especially for verbs derived from nouns and adjectives. For example, the relationship between *služiti* and *službovati* could not be established, because the derivational path looks like this:

služiti 'to serve'$_{\mathrm{V}}$ → *služba* 'service'$_{\mathrm{N}}$ → *službovati* 'being in civil service'$_{\mathrm{V}}$

Although *služiti* and *službovati* share the same root *slug* and belong to the same derivational tree, the two verbs are derived from different stems: *služ-* (comprised only of the root *slug*) and *služb-* (comprised of the root *slug* and the nominal suffix *-b-*). This and other problems in terms of the limitations of the data model were tackled in Štefanec et al. (2013).

The new data model is a combination of principles taken from the previous models and new ones gained from detailed analysis of data as described in this paper. The description of word-formation properties is stored separately from the morphological structure of lexemes whereas derivational connections between them are explicitly created. We believe that this model has enough descriptive power to accommodate and describe the entire Croatian lexicon.

4.1. The new CroDeriv data model

In the new model, following the theoretical approach to the morphological analysis presented in Section 3.1, the lexemes are analyzed for morphemes. Technically, the morphemes are presented as sequences of characters (empty sequences corresponding to zero-morphs are also possible). These sequences at the surface layer are identified as allomorphs and connected to their respective morphemes at the deep layer.

The word-formation description in the data model is presented in the form of building blocks called clusters. Clusters are multi-morphemic units that reflect word-formation processes and roughly correspond to stems/affixes, as presented in Section 3.2.1. The only difference between them is that suffixal clusters do not contain inflectional suffix, which is stored separately. Further, there are no discontinuous clusters. This means that simultaneous prefixation and suffixation is based on simultaneous adding of two types of clusters.

The new design of the database is capable of dealing with compound lexemes by the introduction of the notion of compounding segments. Compound lexemes are split into two or more compounding segments, where the compounding segment on the left side consists of the stem and the interfix, while compounding segment on the right contains the other stem and suffixes. E.g., the compound lexeme *knjigovežnica* 'bindary' is split into two compounding segments: *knjigo* + *vežnica*. Compounds consisting of more than two compounding segments are split as follows:

starocrkvenoslavenski 'Old Church Slavonic' = *staro* + *crkveno* + *slavenski*.
Compounding segments are objects that can be connected to other lexemes by derivational links. If two or more compounding segments can be identified in a lexeme, this lexeme has more than one parent in the database structure. However, due to complexity problem of querying graphs, only one connection will be marked as primary to keep the derivational network as a tree-like structure.

4.2. Technical solutions

The new CroDeriv system is a database-driven server application, developed in Django, a high-level Python web-framework[28] with Django REST framework toolkit[29]. The application supports data querying and retrieval via REST over HTTP. Default data retrieval format is JSON. UDer format[30] is also supported, where applicable, and it can be requested by the client using content negotiation principles.

Data is stored in a PostgreSQL relational database in a normalized form. Since graph-like structures are extremely expensive to query, PostgreSQL Materialized Views were used to increase time efficiency. Materialized Views, as normal Views, use the database rule system, but their result persists in a table-like form until refreshed. This means that highly complex data structures can be transformed in a way which is more redundant but facilitates easy querying, and that this time-expensive operation of transforming will be done sufficiently rarely, probably only after some content is added or changed. In the views, the lexemes' morphological and word-formation structures were pre-computed into easily searchable representations, and paths to every node (i.e. lexeme) in the graph were linearized and stored in a flattened form. On top of that, indexes were added to all searchable fields in the view, which resulted in significant improvement in search latency.

4.3. Querying the CroDeriv database

Beside the possibility to search the database using simple queries, which is the option interesting mostly to the general public, a simple query language, similar to corpus query language (CQL), was constructed which will enable more complex and refine queries.

CroDeriv system supports two general types of queries: lexeme-structural and tree-predecessor. The first type searches for lexemes with particular morphological or word-formation structure. For example, query

```
[prefix="pre"]
```

would return all lexemes starting with a prefix *pre-*. Also, query

[28] `https://www.djangoproject.com/`

[29] `https://www.django-rest-framework.org/`

[30] See Vidra et al. (2019) for detailed description of the format.

```
[morpheme=".+"]*[root="pis"]
```

would return all lexemes that contain the root *pis*. Similarly, when searching for lexemes with particular word-formation pattern, query

```
[prefix="pre"]
```

would return all lexemes that were derived from another lexeme by means of prefixing with *pre-*. Also,

```
[cluster=".+"]*[suffix="inj"]
```

would return all lexemes that were derived with a suffixing word-formation element *-inj-*. It is important to notice that lexeme-structural search queries match results always from the beginning.

The second type of search searches for lexemes in a particular derivational path. For example, query

```
{pos="A"}{pos="N"}{pos="V"}
```

would return all verbs derived from nouns, which were derived from adjectives. Also, query

```
{aspect="biaspectual"}{reflexivity="reflexive"}
```

would return all reflexive verbs derived from biaspectual verbs. This type of queries matches results from the end, i.e. it is possible to search only up the derivational tree, and not down.

Finally, the two types of queries can also be combined. For example, query

```
{aspect="biaspectual"}{aspect="perfective",
    morpho=[morpheme=".+"]*[root="ču"]}
```

would return all perfective verbs that contain the root *ču* and are derived from biaspectual verbs.

5. Concluding remarks and future work

In this paper, we presented the design of CroDeriv 2.0 and its online search interface, required to include non-verbal lemmas as well as to present various derivational properties and relations of Croatian lexemes. CroDeriv 2.0 is designed to comprise the information about morphological structures, word-formation patterns, and derivational relations among Croatian lexemes. We believe that additional information provided for each lemma, e.g. about grammatical categories or external links to online dictionaries, will make this lexicon even more attractive to users.

As mentioned, we intend to use manually analyzed material to build an automatic procedure for morphological and word-formation analysis. This will facilitate the analysis of new lemmas and their inclusion in the lexicon.

Acknowledgements

This research is supported by the short-term support of the University of Zagreb.

Bibliography

Aronoff, Mark and Kristen Fudeman. *What is Morphology. Second Edition*. Wiley-Blackwell, Chichester, 2011.

Babić, Stjepan. *Tvorba riječi u hrvatskome književnome jeziku*. Hrvatska akademija znanosti i umjetnosti : Globus, Zagreb, 2002.

Bagasheva, Alexandra. Comparative semantic concepts in affixation. In Santana-Lario, Juan and Salvador Valera-Hernández, editors, *Competing Patterns in English Affixation*, Linguistic Insights, pages 33–66. Peter Lang, Bern : Berlin : Bruxelles : Frankfurt am Main : New York : Oxford : Wien, 2017.

Bajestan, Elnaz Shafaei, Diego Frassinelli, Gabriella Lapesa, and Sebastian Padó. DErivCelex: Development and Evaluation of a German Derivational Morphology Lexicon based on CELEX. In *Proceedings of the Workshop on Resources and Tools for Derivational Morphology* (*DeriMo*), pages 117–127, Milano, 2017. EDUCatt.

Barić, Eugenija, Mijo Lončarić, Dragica Malić, Slavko Pavešić, Mirko Peti, Vesna Zečević, and Marija Znika. *Hrvatska gramatika*. Školska knjiga, Zagreb, 1995.

Filko, Matea. *Unutarleksičke i međuleksičke strukture imeničkoga dijela hrvatskoga leksika* (*Intralexical and Interlexical Structures of the Nominal Part of the Croatian Lexicon*). Phd thesis, Faculty of Humanities and Social Sciences, University of Zagreb, Zagreb, 2020.

Filko, Matea, Krešimir Šojat, and Vanja Štefanec. Redesign of the Croatian derivational lexicon. In Žabokrtský, Zdeněk, Magda Ševčíková, Eleonora Litta, and Marco Passarotti, editors, *Proceedings of the Second International Workshop on Resources and Tools for Derivational Morphology*, pages 71–80, Prague, 2019. Charles University.

Filko, Matea and Krešimir Šojat. Expansion of the Derivational Database for Croatian. In Litta, Eleonora and Marco Passarotti, editors, *Proceedings of the Workshop on Resources and Tools for Derivational Morphology* (*DeriMo*), pages 27–37, Milan, 2017. EDUCatt.

Gaeta, Livio and Davide Ricca. Frequency and productivity in Italian derivation: A comparison between corpus-based and lexicographical data. *Italian Journal of Linguistics / Rivista di Linguistica*, 15(1):63–98, 2003.

Habash, Nizar and Bonnie Dorr. A categorial variation database for English. In *Proceedings of NAACL-HLT*, pages 17–23, Edmonton, 2003. AL. doi: 10.3115/1073445.1073458.

Hathout, Nabil and Fiammetta Namer. Démonette, a French derivational morpho-semantic network. *Linguistic Issues in Language Technology*, 11(5):125–168, 2014.

Kyjánek, Lukáš, Zdeněk Žabokrtský, Magda Ševčíková, and Jonáš Vidra. Universal Derivations Kickoff: A Collection of Harmonized Derivational Resources for Eleven Languages. In Žabokrtský, Zdeněk, Magda Ševčíková, Eleonora Litta, and Marco Passarotti, editors, *Proceedings of the Second International Workshop on Resources and Tools for Derivational Morphology*, pages 101–110, Prague, 2019. Charles University.

Körtvélyessy, Lívia. Cross-linguistic research into derivational networks. In Žabokrtský, Zdeněk, Magda Ševčíková, Eleonora Litta, and Marco Passarotti, editors, *Proceedings of the Second International Workshop on Resources and Tools for Derivational Morphology*, pages 1–4, Prague, 2019. Charles University.

Lehrer, Adrienne. Polysemy in derivational affixes. In Nerlich, Brigitte, Zazie Todd, Vimala Herman, and David D. Clarke, editors, *Polysemy. Flexible Patterns of Meaning in Mind and Language*, pages 218–232. De Gruyter Mouton, New York, 2003. doi: 10.1515/9783110895698.217.

Lieber, Rochelle. *Morphology and lexical semantics*. Cambridge University Press, New York, 2004. doi: 10.1017/CBO9780511486296.

Lieber, Rochelle. *Introducing Morphology*. Cambridge University Press, New York, 2009. doi: 10.1017/CBO9781316156254.

Litta, Eleonora, Marco Passarotti, and Chris Culy. Formatio formosa est. Building a Word Formation Lexicon for Latin. In *Proceedings of the Third Italian Conference on Computational Linguistics (CLiC-it 2016)*, pages 185–189, Napoli, 2016. Accademia University Press. doi: 10.4000/books.aaccademia.1799.

Ljubešić, Nikola and Filip Klubička. {bs,hr,sr}WaC - Web corpora of Bosnian, Croatian and Serbian. In *Proceedings of the 9th Web as Corpus Workshop (WaC-9)*, pages 29–35, Gothenburg, 2014. Association for Computational Linguistics.

Manova, Stela. Affix Order and the Structure of the Slavic Word. In Manova, Stela, editor, *Affix Ordering Across Languages and Frameworks*, pages 205–230. Oxford University Press, January 2015. doi: 10.1093/acprof:oso/9780190210434.003.0009.

Marković, Ivan. *Uvod u jezičnu morfologiju*. Number 6 in Biblioteka Thesaurus. Disput, Zagreb, 2012.

Marković, Ivan. *Hrvatska morfonologija*. Number 7 in Biblioteka Thesaurus. Disput, Zagreb, 2013.

Moguš, Milan, Maja Bratanić, and Marko Tadić. *Hrvatski čestotni rječnik*. Školska knjiga : Zavod za lingvistiku Filozofskoga fakulteta, Zagreb, 1999.

Pala, Karel and Pavel Šmerk. Derivancze — Derivational Analyzer of Czech. In Král, Pavel and Václav Matoušek, editors, *Text, Speech, and Dialogue: 18th International Conference, TSD 2015*, pages 515–523, Berlin: Heidelberg, 2015. Springer. doi: 10.1007/978-3-319-24033-6_58.

Passarotti, Marco and Francesco Mambrini. First Steps towards the Semi-automatic Development of a Wordformation-based Lexicon of Latin. In Calzolari, Nicoletta et al., editor, *Proceedings of the Eighth International Conference on Language Resources and Evaluation*, pages 852–859, Istanbul, 2012. ELRA.

Silić, Josip and Ivo Pranjković. *Gramatika hrvatskoga jezika: za gimnazije i visoka učilišta*. Školska knjiga, Zagreb, 2005.

Stewart, Thomas W. *Contemporary Morphological Theories. A User's Guide*. Edinburgh University Press, Edinburgh, 2016.

Tadić, Marko. New version of the Croatian National Corpus. In Hlaváčková, Dana, Aleš Horák, Klara Osolsobě, and Pavel Rychlý, editors, *After Half a Century of Slavonic Natural Language Processing*, pages 199–205. Masaryk University, Brno, 2009.

Talamo, Luigi, Chiara Celata, and Pier Marco Bertinetto. DerIvaTario: An annotated lexicon of Italian derivatives. *Word Structure*, 9(1):72–102, 2016. doi: 10.3366/word.2016.0087.

Vidra, Jonáš, Zdeněk Žabokrtský, Magda Ševčíková, and Lukáš Kyjánek. DeriNet 2.0: Towards and All-in-One Word-Formation Resource. In Žabokrtský, Zdeněk, Magda Ševčíková, Eleonora Litta, and Marco Passarotti, editors, *Proceedings of the Second International Workshop on Resources and Tools for Derivational Morphology*, pages 81–89, Prague, 2019. Charles University.

Zeller, Britta, Jan Šnajder, and Sebastian Padó. DErivBase: Inducing and Evaluating a Derivational Morphology Resource for German. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics* (*Volume 1: Long Papers*), pages 1201–1211, Sofia, 2013. Association for Computational Linguistics.

Ševčíková, Magda and Zdeněk Žabokrtský. Word-Formation Network for Czech. In Calzolari, Nicoletta et al., editor, *Proceedings of the Ninth International Conference on Language Resources and Evaluation*, pages 1088–1093, Reykjavik, 2014. ELRA.

Šnajder, Jan. DERIVBASE.HR: A High-Coverage Derivational Morphology Resource for Croatian. In Calzolari, Nicoletta et al., editor, *Proceedings of the 9th Language Resources and Evaluation Conference* (*LREC 2014*), pages 3371–3377, Reykjavik, 2014. ELRA.

Šojat, Krešimir, Matea Srebačić, and Marko Tadić. Derivational and Semantic Relations of Croatian Verbs. *Journal of Language Modelling*, 0(1):111, 2012. doi: 10.15398/jlm.v0i1.34. URL `http://jlm.ipipan.waw.pl/index.php/JLM/article/view/34`.

Šojat, Krešimir, Matea Srebačić, and Vanja Štefanec. CroDeriV i morfološka raščlamba hrvatskoga glagola. *Suvremena lingvistika*, 75:75–96, 2013.

Šojat, Krešimir, Matea Srebačić, and Tin Pavelić. CroDeriV 2.0.: Initial Experiments. In Przepiórkowski, Adam and Maciej Ogrodniczuk, editors, *Advances in Natural Language Processing*, volume 8686, pages 27–33. Springer International Publishing, Cham, 2014. doi: 10.1007/978-3-319-10888-9_3. URL `http://link.springer.com/10.1007/978-3-319-10888-9_3`.

Štefanec, Vanja, Krešimir Šojat, and Matea Srebačić. A Method for the Computational Representation of Croatian Morphology. In Kłopotek, M. A. et al., editor, *Language Processing and Intelligent Information Systems*, pages 80–91. Springer, 2013. doi: 10.1007/978-3-642-38634-3_10.

Žabokrtský, Zdeněk, Magda Ševčíková, Milan Straka, Jonáš Vidra, and Adéla Limburská. Merging Data Resources for Inflectional and Derivational Morphology in Czech. In Calzolari, Nicoletta et al., editor, *Proceedings of the Tenth International Conference on Language Resources and Evaluation*, pages 1307–1314, Portorož, 2016. ELRA.

Address for correspondence:
Matea Filko
`matea.filko@ffzg.hr`
Faculty of Humanities and Social Sciences, University of Zagreb
Ivana Lučića 3, 10000 Zagreb, Croatia

PBML

The Prague Bulletin of Mathematical Linguistics
NUMBER 115 OCTOBER 2020 105–127

Morphological Networks for Persian and Turkish: What Can Be Induced from Morpheme Segmentation?

Hamid Haghdoost,[a] Ebrahim Ansari,[ab] Zdeněk Žabokrtský,[b]
Mahshid Nikravesh,[a] Mohammad Mahmoudi[a]

[a] Department of Computer Science and Information Technology,
Institute for Advanced Studies in Basic Sciences
[b] Institute of Formal and Applied Linguistics,
Faculty of Mathematics and Physics, Charles University

Abstract

In this work, we propose an algorithm that induces morphological networks for Persian and Turkish. The algorithm uses morpheme-segmented lexicons for the two languages. The resulting networks capture both derivational and inflectional relations. The network induction algorithm can use either manually annotated lists of roots and affixes, or simple heuristics to distinguish roots from affixes. We evaluate both variants empirically. We use our large hand-segmented set of word forms in the experiments with Persian, which is contrasted with employing only a very limited manually segmented lexicon for Turkish that existed previously. The network-induction algorithm uses gold segmentation data for initializing the networks, which are subsequently extended with additional corpus-attested word forms that were unseen in the segmented data. For this purpose, we use existing morpheme-segmentation tools, namely supervised and unsupervised version of Morfessor, and (unsupervised) MorphSyn. The experimental results show that the accuracy of segmented initial data influences derivational network quality.

1. Introduction

Even though the Natural Language community put more focus on inflectional morphology in the past, one can observe a growing interest in research on derivational morphology (and other aspects of word formation) recently, leading to the existence of various morphological data resources. One relatively novel type of resource

 Corresponding author: ansari@iasbs.ac.ir
Cite as: Hamid Haghdoost, Ebrahim Ansari, Zdeněk Žabokrtský, Mahshid Nikravesh, Mohammad Mahmoudi. Morphological Networks for Persian and Turkish: What Can Be Induced from Morpheme Segmentation?. The Prague Bulletin of Mathematical Linguistics No. 115, 2020, pp. 105–127. doi: 10.14712/00326585.007.

is word formation networks, some of which represent information about derivational morphology in the shape of a rooted tree. In such networks, the derivational relations are represented as directed edges between nodes that represent lexemes (Lango et al., 2018).

In our work, we present a procedure that builds a morphological network for Persian and Turkish using a word segmentation lexicon. The resulting network (a directed graph) represents each cluster of morphologically related word forms as a tree-shaped component of the overall graph. Thus, the specific feature of our network is that it captures both derivational and inflectional relations in a single structure (at this moment, the two types of relations are not distinguished at all). Figure 1 shows an example of such a tree for the Persian language, which represents a base morpheme meaning "to know" and all its derived and inflected descendants. In this example, the path from the root to one of the deepest leaves corresponds to the following meanings: (1) "to know", (2) "knowledge"/"science", (3) "university", (4) "a person from a university", (5) "some people from a university".

What we use as a primary source of morphological information for Persian is our manually annotated morpheme-segmented lexicon of Persian word forms, which is the only segmented lexicon for this language. At the same time, to the best of our knowledge, this lexicon containing 45,300 words could be considered as the biggest publicly available manually segmented lexicon at all (for any language). For Turkish, we use a previously existing morpheme-segmented dataset published in the Morpho Challenge 2010 Shared Task[1]. It has about 600K unsegmented words and 1000 gold standard segmented words.

Additional corpus-attested words that are not stored in the manually annotated lexicon are added into the network using automatic morpheme-segmentation methods. In order to segment new words, we used both supervised and unsupervised versions of Morfessor (Creutz et al., 2007; Grönroos et al., 2014), a popular automatic segmentation toolkit, and the MIT Arabic Segmenter (Lee et al., 2011). After performing the segmentation of unseen word forms, the process of inducing morphological relations is the same as for hand-segmented words.

The paper is organized as follows: Section 2 addresses related work on derivational morphology networks and morphological segmentation. Section 3 describes our morpheme-segmented Persian lexicon, including details on technical preprocessing and manual annotation. Section 4 describes our network construction approach. Section 5 presents experimental results and a discussion of various experiment configurations. Finally, Section 6 concludes.

[1] http://morpho.aalto.fi/events/morphochallenge2010/datasets.shtml

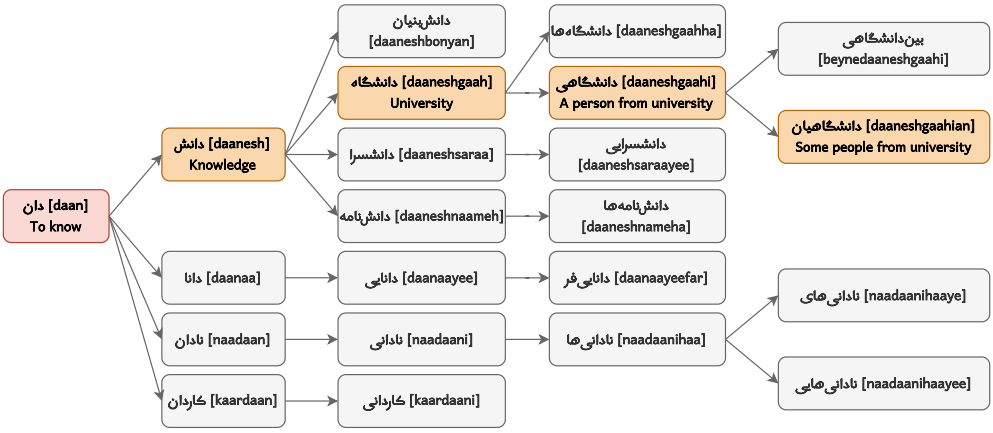


Figure 1. A sample of a Persian morphological tree for root دان [dan] which means "to know". The path from the root to one of the deepest leaf corresponds to the following meanings: (1) "to know", (2) "knowledge"/"science", (3) "university", (4) "a person from university", (5) "some people from university".

2. Related work

For some languages, intensive research exists with a focus on the construction of resources specializing in derivation. For instance, DerivBase (Zeller et al., 2013) describes a rule-based framework for inducing derivational families in German, and DErivCelex (Shafaei et al., 2017) presents an algorithm that extracts derivationally related lexicons for this language too. Hathout and Namer (2014) proposed Démonette that offers derivational morpho-semantic information for French. Šnajder (2014) presented DerivBase.Hr as a high-coverage derivational morphology resource for Croatian. Another derivational resource for Croatian is CroDeriV presented by Šojat et al. (2014) that contains data about the morphological structure and derivational relatedness of verbs. The Derinet network for Czech (Ševčíková and Žabokrtský, 2014; Žabokrtský et al., 2016) is a large linguistic resource containing over 1 million lexemes. Rafea and Shaalan (1993) presented a lexical analyzer for inflected Arabic words. For the English language, Habash and Dorr (2003) constructed and evaluated a large-scale database called CatVar, which contains categorical variations of English lexemes. Other relevant resources are (Vilares et al., 2001; Baranes and Sagot, 2014; Lango et al., 2018) for Spanish, Word Formation Latin (Litta et al., 2016), and (Piasecki et al., 2012; Kaleta, 2017; Lango et al., 2018) for Polish. Cross-linguistic research into morphological derivations is described in Kőrtvélyessy (2019). Kyjánek et al. (2019) presented

an attempt at collecting the existing derivational resources for eleven languages and at harmonizing them under a unified annotation scheme.

However, for many other languages, the data resources which provide information about derived words are scarce or lacking.

Our study is focused on Persian and Turkish. Both languages are morphologically rich languages with powerful and versatile word formation processes.

Persian is a Western Iranian language belonging to the Indo-European languages, predominantly spoken within Iran, Afghanistan and Tajikistan. Having many affixes to form new words (a few hundred), the Persian language uses derivational agglutination to form new words from nouns, adjectives, and verb stems.

Turkish is the major member of the Turkic language family, which is a subfamily of the Altaic languages. Turkish is spoken in Turkey, Cyprus, and elsewhere in Europe and the Middle East. Extensive agglutination is a prominent feature of both the Turkish language and the Persian language.

To our knowledge, research on Persian morphology is very limited. Rasooli et al. (2013) claimed that performing morphological segmentation in the pre-processing phase of statistical machine translation could improve the quality of translations for morphologically rich and complex languages. Although they segmented only an extremely limited and non-representative sample of Persian words (tens of Persian verbs), the quality of their machine translation system increases by 1.9 points of BLEU score. Arabsorkhi and Shamsfard (2006) proposed an algorithm based on Minimum Description Length with certain improvements for discovering the morphemes of the Persian language through automatic analysis of corpora. However, since no Persian segmentation lexicon was made publicly available, we decided to create a manually segmented lexicon for Persian that contains 45K words now.

For our approach, we also need automatic morpheme segmentation. The discussion about this task can be traced back to Harris (1955). Recent research on morpheme segmentation has been usually focused on unsupervised learning (Goldsmith, 2001; Creutz and Lagus, 2002; Poon et al., 2009; Narasimhan et al., 2015; Cao and Rei, 2016), whose goal is to find the segmentation boundaries using an unlabeled set of word forms (or possibly a corpus too). Probably the most popular unsupervised systems are LINGUISTICA (Goldsmith, 2001) and Morfessor, with a number of variants (Creutz and Lagus, 2002; Creutz et al., 2007; Grönroos et al., 2014); a semi-supervised extension of Morfessor was introduced by Kohonen et al. (2010). Poon et al. (2009) presented a log-linear model that uses overlapping features for unsupervised morphological segmentation. Lee et al. (2011) describe the MIT Arabic Segmenter, which uses the syntactic context of words and utilizes connections between part-of-speech categories and morphological segmentation of words. Narasimhan et al. (2015) proposed Morphochain, which is an unsupervised morphological analysis model integrating orthographic and semantic perspectives.

In our study, we use a combination of hand-annotated segmentation with segmentation generated by the supervised version of Morfessor, whose performance for our

purposes is superior to the performance of the unsupervised version, as described in Section 4.3.

3. Data

In this section we introduce the data which we used in our work. Section 3.1 describes the Persian data and Section 3.2 is a brief description of the Turkish corpus. We do not provide details about morphology of Persian and Turkish; they can be found in Jones (1807), and in Underhill (1976), respectively.

3.1. Data for Persian

We have introduced a hand-annotated segmentation data for Persian formerly (Haghdoost et al., 2019). In the following text, we describe the procedure of data creation in more detailed specification against the previous paper.

3.1.1. Corpus Collection

For compiling a set of word forms to be covered by our network, we use three Persian corpora focused on different domains.

The first source is the Persian Wikipedia (Karimi et al., 2018). The data is extracted from the Wikipedia archive that is available from the Linguatools website.[2] The files provided in the Wikipedia dataset are stored in an XML file format containing all the documents in Wikipedia for many languages, out of which we use only the Persian part. We removed XML markup and used only plain texts from the corpus.

The second source is the Bijankhan corpus (Bijankhan et al., 2011), which is a popular Persian monolingual corpus. The corpus collects daily news and other texts. The Bijankhan collection contains about 2.6 million words manually tagged with a tag set that contains 40 Persian POS tags. Again, we used only plain texts from this corpus.

The third language resource that we used is Persian-NER[3] (Poostchi et al., 2018), developed for the task of Persian named entity recognition. The resource recognizes named entities such as persons, places, and organizations.

3.1.2. Preprocessing and Tokenization

We extracted and normalized Persian sentences from all three corpora using the **Hazm** toolkit.[4] Hazm is a Python library for processing Persian text, including tokenization and lemmatization.

[2] https://linguatools.org/tools/corpora/wikipedia-monolingual-corpora/

[3] https://github.com/HaniehP/PersianNER

[4] https://github.com/sobhe/hazm

For the stemming task, we used the stemmer tool presented by Taghizadeh (Taghi-Zadeh et al., 2015). This stemmer combines cues related to orthography, corpus frequency, and syntactic distributions to induce stemming rules. It processes data in two steps. In the first step, all words of the annotated text corpus are used to automatically induce stemming rules. In the second part, the rule-based stemmer uses those stemming rules to induce words' stems. For lemmatization, we used a Persian lemma collection and the mentioned tool.

An important feature of the written form of Persian and Arabic languages is the existence of semi-space. A semi-space separates neighboring parts of a word and the separated character is narrower than a normal space. It prevents sticking morphemes. For example, word "کتابها" (books) is a combination of the word "کتاب" and "ها", in which the former is Persian translation of word "book" and the latter is the morpheme for a plural form. We can say these semi-space signs segment words into smaller morphemes. However, in formal writing and in all normal Persian corpora, this space is neglected frequently and it could make a lot of problems in Persian and Arabic morphological segmentation tasks. For example both forms for the previous example, "کتاب ها" and "کتابها", are considered correct in Persian texts and have the same meaning. In this work, all missing semi-spaces are automatically detected and corresponding words are updated accordingly.

Some words in the included corpora cannot be considered correct Persian words. To reduce the number of such words, we decided to remove words with low frequency. Words with more than 10 occurrences in the corpora were selected for manual annotation and those having less than ten occurrences were ignored in our experiments. Selected words were stored in a spreadsheet table, as illustrated in Figure 2.

3.1.3. Manual Annotation

We distributed 80K words resulting from the previous phase among our sixteen annotators in such a way that each word was annotated by two independent annotators. Annotators decided about the lemma of a word under question, segmentation points, plurality, ambiguity (whether a word has more than one meaning), being Named Entity, or they might mark the word for deletion if they think it is not a proper Persian word. Denoting the segmentation points was sufficient for generating derivational network. However, we decided to extract more information about words, because denoting the other pieces of information was not so time consuming, and they could be useful in future work. For example, denoting Named Entities could be utilized in Named Entity Recognition tasks. Our automatic segmenter tool is based on the work of Taghi-Zadeh et al. (2015), in which suffixes are stripped using rules automatically induced from a corpus. The segmenter offered a pre-segmentation (i.e., some very simple suggestions) to our annotators if it finds a word's segmentation reaching a high confidence score.

50165	X	شیدا	شیدایی	0		126	ش	ی	د	ا	ی	ی		
50166	X	شیده	شیده	0	N	17	ش	ی	د	ه				
50167	X	شیدور	شیدور	0		15	ش	ی	د	و	ر			
50168	X	شیدی	شیدی	0	N	43	ش	ی	د	ی				
50169	X	شیر	شیر	1	N	4694	ش	ی	ر					
50170	X	شیرآباد	شیرآباد	0		21	ش	ی	ر	آ	ب	ا	د	
50171	X	شیرآبه	شیرآبه	0		12	ش	ی	ر	آ	ب	ه		
50172	X	شیرآلات	شیرآلات	0		27	ش	ی	ر	آ	ل	ا	ت	
50173	X	شیرا	شیرا	0	N	23	ش	ی	ر	ا				
50174	X	شیرابه	شیرابه	0		32	ش	ی	ر	ا	ب	ه		
50175	X	شیراز	شیراز	0	N	6953	ش	ی	ر	ا	ز			
50176	X	شیرازه	شیرازه	0		56	ش	ی	ر	ا	ز	ه		

Figure 2. A snapshot of extracted data stored in a spreadsheet editor. Column 1: row ID. Column 2: distinguishing proper Persian words (“X”) from words to be deleted (“D”). Column 3: the stem of the word. Column 4: the original word form. Column 5: marking ambiguous words, 0 means “non-ambiguous” and 1 denotes “ambiguous”. Column 6: The word is a Named Entity (N) or not (empty). Column 7: frequency of the word in the source corpus. Rest: individual characters in the word form.

word	lemma	form	ambiguity	segmentation			
آزمایش (experiment)	آزمایش	X	0	آزما	یش		
آزمایشات (experiments)	آزمایش	X	0	آزما	یش	ات	
آزمایشاتی (some experiments)	آزمایش	X	0	آزما	یش	ات	ی
آسیه (Asieh (NE))	آسیه	E	0	آسیه			
دام (trap - livestock)	دام	X	1	دام			

Figure 3. A sample of hand-annotated dataset.

We removed almost words that were marked to be deleted by both annotators. The remaining 50K words (including around 12K words, for which the annotator delivered completely identical annotation) were sent for the inter-annotation disagreement resolution. In this phase, all disagreements were resolved. Finally, all words were quickly reviewed by two Persian linguists. The whole process took almost six weeks and the total number of words stored in the resulting lexicon is about 45K. Lemmas and some extra information about those words are also included.

In the final released dataset, every word is formatted as follows: Words are separated by "\n" and in each line (for each word) we have this information:

$word\ lemma\ form\ ambiguity\ segment_1\ segment_2\ ...\ segment_n$

Where "form" could be one of these:
- V: Verb
- E: Named entity word
- I: Irregular plural
- X: None of the above

The "ambiguity" field could be 0 which means the word has only one meaning and is 1 when the word has more than one meaning. Figure 3 shows a sample of final annotated data.

The resulting data resource (Ansari et al., 2019a) is publicly available under a permissive license (CC BY-NC-SA) for other researchers interested in the morpheme segmentation of Persian. Recently, we used the data for supervised morpheme segmentation task (Ansari et al., 2019b).

3.2. Data for Turkish

We have used a text corpus for Turkish that is publicly available from the Morpho Challenge 2010 event, whose aim was to find the morpheme analysis of the word forms in the data. There was a small set of gold-standard segmented data provided for semi-supervised learning of morpheme analysis, and we have used it in our supervised segmentations. In the mentioned dataset there is a list of word forms which is extracted from a text corpus and each word in the list is preceded by its frequency in the corpus used. The corpora have been preprocessed for the Morpho Challenge (tokenized, lower-cased, some conversion of character encodings).

The format of gold segmented data as well as the output of the mentioned input data for Morpho Challenge is like this: Each line of the file contains a word (e.g., "kontrole") separated from its analysis (e.g., "kontrol +DAT") by one TAB character. Morpheme labels in the analysis are separated from each other by a space character. For some words there are multiple correct analyses. These alternative analyses are separated by a comma (,). The Turkish gold-standard analyses have been obtained

from a morphological parser developed at Boğaziçi University (Sak et al., 2008). It is based on Oflazer's finite-state machines (Oflazer, 1994), with a number of changes.

4. Morphological Network Construction

In this section, the network induction based on a set of morpheme-segmented word forms is described. Subsection 4.1 introduces our algorithm developed for this task, while Subsection 4.2 describes an extension employing automatic segmentation. Subsection 4.3 describes an automatic network expansion procedure using a morphological segmenter named Morfessor and Finally in Subsection 4.4, the effect of segmentation algorithm is examined on two languages.

4.1. Automatic Network Construction

The core idea of this work is to construct a morphological network using a morpheme-segmented lexicon, be the segmentation loaded from a human-annotated lexicon, or automatically in a fully unsupervised or semi-supervised fashion.

In our proposed algorithm, first, we partition the set of word forms into subsets sharing the same root morphemes, and thus the root morpheme must be recognized among all morphemes in a given word form. We approximate the distinction between root morphemes and affixes using the number of occurrences of individual morphemes in the lexicon. After gathering the frequency counts, the m most frequent segments (we used 100 and 200 for m in our experiments) are removed from the set of potential root morphemes; all the remaining morphemes are stored in a set named $roots$. The underlying intuition is that affixes tend to repeat across many derivational clusters, and thus tend to be more frequent than root morphemes.[5] Table 1 shows an example of the most frequent segments based on our Persian segmented lexicon; all of them are classified correctly using this heuristics (i.e., all of them serve as affixes in Persian).

In the second phase, we add nodes to our morphological graph (i.e., the network contains morphological trees) based on the assembled set of root morphemes. For each r_i from the $roots$ set, we create a set of words that contain r_i. We name this set $words_i$. Now, we add r_i as a new node to our derivational graph. In the next step, we find and connect all the words in $words_i$ in the network. We divide all the words in $words_i$ into n smaller sets $words_{i,2}, words_{i,3}, ..., words_{i,n}$ based on the number of their segments. The set $words_{i,j}$ includes all words containing r_i and their number of segments is equal to j. First, we check all w in $words_{i,2}$ and if it contains a node in the tree that includes r_i, we add it to the network graph, otherwise we add w to the $remaining$ set. Then, for the next group, $words_{i,3}$, we follow a similar procedure;

[5] For simplicity of the model, we assume the boundary between root morphemes and affixes to be sharp. We do not introduce any borderline category such as affixoids.

however, we add all w in $words_{i,3}$ when it contains a node existing in $words_{i,2}$ (i.e., set of words with two segments). Then we add them to $remaining$ if there is not any subset in our current graph. We iterate this procedure until we pass all sets. Now, for each w in $remaining$ set, we check all added nodes and add w as a child of any node with the maximum number of segments. It means it would be connected to the root if there is no other option available.

Algorithm 1 shows a simple pseudocode of the segmentation graph generating procedure. The $generate$ function is recursive and gets $root$, current $tree$, remaining $words$ and current $step$ as the input parameters and returns a new $tree$ and remaining $words$. The $overlap$ function gets two words as the input and checks the left and right overlap count of the morphemes and returns the maximum of them.

For example, consider two words "understanding" with segments "understand+ing" and "misunderstanding" with segments "mis+understand+ing". The left overlap number of these words is 0 because there are not equal segments from the starting point of words but the right overlap number of them is 2 because two segments (understand and ing) are equal when we are browsing segments from the reverse side, from end to beginning. Finally, the algorithm returns the maximum number of them as a return value.

Algorithm 1 pseudocode of generating derivational graphs.

```
1:  function GENERATE(root, tree, words, n)          ▷ recursive network generation
2:      tree[root] ← root
3:      for all words do
4:          for all leaves(tree[root]) do
5:              if OVERLAP(leaf, word) > n then
6:                  setChildToLeaf(tree, leaf, word)
7:              else
8:                  appendTo(remains, word)
9:              for all leaves do
10:                 tree, remains ← GENERATE(leaf, tree, remains, n + 1)
11:     return tree, remains
12: function OVERLAP(x, y)
13:     return max(leftOverlap(x, y), rightOverlap(x, y))
14: for all segmentationSets do
15:     tree, remains ← GENERATE(root, {}, set, 1)
```

4.2. Semi-automatic Network Construction

As expected, our frequency-based identification of root morphemes vs. affixes is only an approximation; there are frequent morphemes such as [shah] "king" (clearly

not an affix) among the first 200 frequent segments. In order to quantify the influence of such wrongly classified affixes, we performed a modified version of the above-described experiment. This time, after frequency counting, we selected the m most frequent morphemes, and one annotator decided whether the morphemes are root morphemes or not (such annotation is not a time-consuming task for a human at all). The rest of the experiment remained the same. Again, we set m equal to 100 or 200.

i	seg.	freq.	i	seg.	freq.	i	seg.	freq.	i	seg.	freq.
1	ی [y]	9118	11	ایی [ee]	583	21	هم [ham]	278	31	اَست [ast]	216
2	ها [haa]	4819	12	اَل [al]	561	22	ید [id]	274	32	اَش [ash]	206
3	ه [h]	2898	13	تر [tar]	746	23	آ [aa]	274	33	دَان [daan]	198
4	آن [aan]	1708	14	آت [aat]	425	24	م [m]	267	34	شان [shaan]	193
5	می [mi]	1112	15	ب [b]	422	25	در [dar]	260	35	گاه [gaah]	192
6	یّ [yee]	941	16	ین [een]	396	26	کار [kaar]	258	36	کن [kon]	189
7	ش [sh]	891	17	ده [deh]	383	27	ساز [saaz]	254	37	پر [por]	187
8	ن [n]	864	18	شد [shod]	359	28	دو [do]	241	38	نَا [naa]	178
9	ند [nd]	782	19	دار [daar]	337	29	بر [bar]	239	39	ت [t]	173
10	د [d]	658	20	و [oo]	308	30	گر [gar]	232	40	شاه [shaah]	164

Table 1. 40 most frequent morphemes in the Persian hand-segmented lexicon, most of which are non-roots. For example, the first morpheme is the indefinite article in Persian, the second and fourth morphemes are two different plural suffixes, the third one is a suffix for female form of names, and finally, the last one is used to create the present continuous form.

4.3. Automatic Network Expansion Using Morpheme Segmentation Generated by Morfessor

Relying on the availability of manually annotated morpheme boundaries for each word in the network is clearly a bottleneck. Thus we propose an automatic procedure to expand the existing derivational network by adding selected unseen words into the graph. In other words, once the primary network based on golden annotations is ready, we try to add new words into it using the core algorithm explained in Section 4.1, just that the morpheme segmentation is produced by an automatic tool such as Morfessor. Figure 4 shows the workflow of the segmentation process.

As is shown in Figure 4, Morfessor is used in two different phases. First, in the initial data segmentation to create the primary morphological network. The second

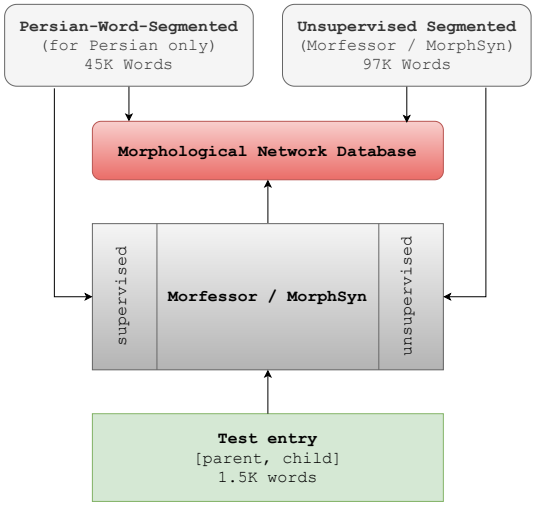


Figure 4. Morphological Network Database construction flowchart which shows the primary network construction and the expansion procedure.

part of adopting Morfessor is when we have some new words (i.e. test words) and we want to add them into our existing network and we can use Morfessor to segment them in an automatic way. In other words, in the testing phase, we have words that do not exist in our hand-annotated dataset and for creating the derivational network of morphemes we need segmentation for them too. We selected Morfessor to segment this new unseen words. Morfessor works in two ways; supervised and unsupervised: we created two models of Morfessor and in the testing phase when a new word is under question, we segment it and add it to our existing tree based on that segmentation.

In this experiment, the unsupervised model is created based on 97K words we collected from the raw text and the supervised Morfessor is trained using the 45K hand-annotated dataset. Experimental results in Section 5 show that the supervised model has better performance in comparison with the unsupervised one in the final tree accuracy.

4.4. The Effect of Segmentation Algorithm

The assumption that the accuracy of derivational networks depends substantially on morpheme segmentation quality is confirmed by another experiment in which we compared derivational networks created using outputs of two different segmentation

Data Language	Segmentation Method	Segmentation Accuracy	Network Accuracy
Persian	Morfessor	0.41	0.856
Persian	MorphSyn	0.37	0.307
Turkish	Morfessor	0.21	0.499
Turkish	MorphSyn	0.12	0.289

Table 2. Better segmentation improves derivational network accuracy.

algorithms. In order to do this, we used Morfessor[6] and MorphSyn[7]. Morfessor is a family of machine-learning methods that segment words into morphemes. More specifically, we used methods named Morfessor Baseline and Categories-MAP; both of which are based on probabilistic generative models. MorphSyn (MIT Arabic Segmenter) is a segmentation tool that uses a connection between part-of-speech categories and morphological properties. Our primary data for both Persian and Turkish was not a text corpus and the context that words occurred in was not denoted in the data. This reason led us to use the first model of MorphSyn. This model is basic and does not model the relationship between words and POS tags.

To compare our selected segmentation algorithms, we trained unsupervised models with 97,000 words of unlabeled data. The derivational network was created using these forms and then the accuracy of the derivational network was reported. For more clarity, we calculated the accuracy of morpheme segmentation and reported it. Table 2 shows the accuracy of unsupervised morphological segmentation and derivational network accuracy for both methods. The reported segmentation accuracy is based on the total word accuracy, i.e. if there was a wrong segmentation boundary on the segmented word, the whole segmentation of word considered as wrong. The segmentation accuracy calculated on a 1000 gold-standard segmented data for both languages and the results are reported. For network accuracy, we selected 400 random parent-child in the trees, then the accuracy is calculated by dividing the count of the correct parent-child in the network, by the total number of selected pairs (400). Morfessor had higher accuracy than MorphSyn as well as in derivational network accuracy that is a result of correct segmentation cases.

5. Experiments and Discussion

In order to estimate the quality of the resulting network, we randomly selected 400 nodes and checked manually if their parent nodes are identified correctly (or if the nodes are correctly marked are derivational tree roots, i.e., they are parentless). We

[6] https://Morfessor.readthedocs.io/en/latest/

[7] http://groups.csail.mit.edu/rbg/code/morphsyn/

Figure 5. Samples of generated trees using our procedures. For example in the left tree, the root of the tree is the word "Square" and the words like "Track and Field" and "Squares" are its first level children.

ran our automatic and semi-automatic versions of the algorithm using two thresholds for skipped root morphemes, 100 and 200. Table 3 summarizes the results for the individual experiment configurations. In all cases, the number of nodes in the generated graphs is 45K, which is equal to the total number of words in our manually segmented lexicon. Finally, Figure 5 shows three sample sub-graphs extracted by our algorithms.

non-root selection	number of non-roots	accuracy
automatic	100	89.5%
automatic	200	86.3%
semi-automatic	100	91.0%
semi-automatic	200	92.8%

Table 3. Accuracy for both automatic and semi-automatic methods using different numbers of non-roots in primary phase on 400 randomly selected nodes (i.e., words).

In the next experiment, we evaluated the expansion of unseen words. Table 4 shows the results of eight configurations of our experiments using Morfessor as the automatic morpheme segmentation tool. In the first half of the table, we used all available words to create out initial network and the unsupervised version of Morfessor is used for the initial segmentation. In the bottom half of Table 4, all rows show the results when the hand-annotated segmented data is used. Similarly to the previous experiment, we removed and cleaned most frequent non-root morphemes in two ways:

in automatic removing during which we ignore all first 200 frequent morphemes, and in manual removing during which the selection and removing is done by an annotator. In other words, the first two columns of this table represent the configuration of the initial tree creation. The third column of Table 4 represents the method we used for segmenting the new words and in this column. Caption "Supervised" declares we used supervised Morfessor, which is trained using 45K hand-annotated data and "Unsupervised" indicates that the segmentation is done by using a fully unsupervised version of Morfessor. For all tests in this experiment, we provided a hand-annotated morphological network with 1500 words.

init. network creation	non-root selection	test words seg.	Accuracy
97K/Segmented by Morfessor	automatic	sup. Morfessor	0.893
97K/Segmented by Morfessor	automatic	uns. Morfessor	0.777
97K/Segmented by Morfessor	manual	sup. Morfessor	0.893
97K/Segmented by Morfessor	manual	uns. Morfessor	0.777
45K Persian-Word-Segmented	automatic	sup. Morfessor	0.919
45K Persian-Word-Segmented	automatic	uns. Morfessor	0.846
45K Persian-Word-Segmented	manual	sup. Morfessor	0.934
45K Persian-Word-Segmented	manual	uns. Morfessor	0.866

Table 4. Accuracy for tree structures on 1.5K unseen words. "test word seg." column indicates the selected algorithm for unseen word segmentation

5.1. Derivation Network for the Turkish Language

Our algorithm relies fundamentally on morpheme segmentation, and the derivational network accuracy is thus directly related to the accuracy of segmentation of data. For more investigation on this claim, we ran our network generating procedure on the Turkish language.

In this experiment, we generated a derivational network from the Turkish part of Morpho Challenge 2010 (Virpioja et al., 2011) dataset.[8] It has about 600K Turkish word forms (i.e., inflected word forms), which are enough for the unsupervised segmentation tasks, and there are 1000 gold-standard segmented words. We used both supervised and unsupervised Morfessor for word segmentation and generated Turkish trees using our network creation algorithm. Table 5 shows the most frequent segments after running unsupervised Morfessor on the data; all of them are suffixes. While the majority of the first 500 most frequent words were suffixes, we decided

[8] http://morpho.aalto.fi/events/morphochallenge2010/datasets.shtml

rank	segment	freq.	rank	segment	freq.	rank	segment	freq.
1	lar	9830	11	larI	5433	21	dan	3865
2	ler	9056	12	da	5330	22	’in	3856
3	n	8779	13	li	5148	23	den	3723
4	i	8729	14	in	5126	24	la	3696
5	a	7906	15	m	5087	25	le	3652
6	e	7749	16	dir	4383	26	I	3631
7	k	6644	17	s	4165	27	larIn	3495
8	de	5982	18	lerin	4024	28	ye	3421
9	leri	5931	19	nin	4009	29	lI	3178
10	si	5456	20	ya	3981	30	lere	2919

Table 5. 30 most frequent morphemes in the Turkish segmented lexicon by unsupervised Morfessor that all of them are suffixes. For example for the 5 first ranked morphemes: “lar” and “ler” are plural suffixes, “n” is possessive suffix for 2nd person singular, “i” is possessive suffix for 3rd person singular, and “a” is equates to “to” or “towards” in English.

to ignore them in the root selection phase using two approaches: automatically and manually.

For evaluation, we created a set of 400 parent-child derivational pairs. In our experiments, we obtained 49.4% accuracy in unsupervised primary data and 47.6% correct parent-child prediction in the best configuration of supervised primary data. The complete evaluation of the Turkish network is presented in Table 6. As is shown in this table, the accuracies shown are not as good as the Persian results; we assume that the most limiting factor is the amount of hand-segmented words for Turkish, which is much smaller than that for Persian.

Primary Data	Remove Count	Network Accuracy
supervised	300	0.476
supervised	500	0.387
unsupervised	300	0.491
unsupervised	500	0.494

Table 6. Accuracy of 400 randomly selected words in the Turkish derivational networks created by supervised (trained on 1000 gold segmented words) and unsupervised (trained by 600k raw words) segmentation algorithms on primary segmented data using two thresholds for non-root removing phase.

Figure 6 shows four samples from the Turkish network created. From left to right they are eteg (Skirt), garib (Strange), kamCi (Whip) and bagaj (Luggage).

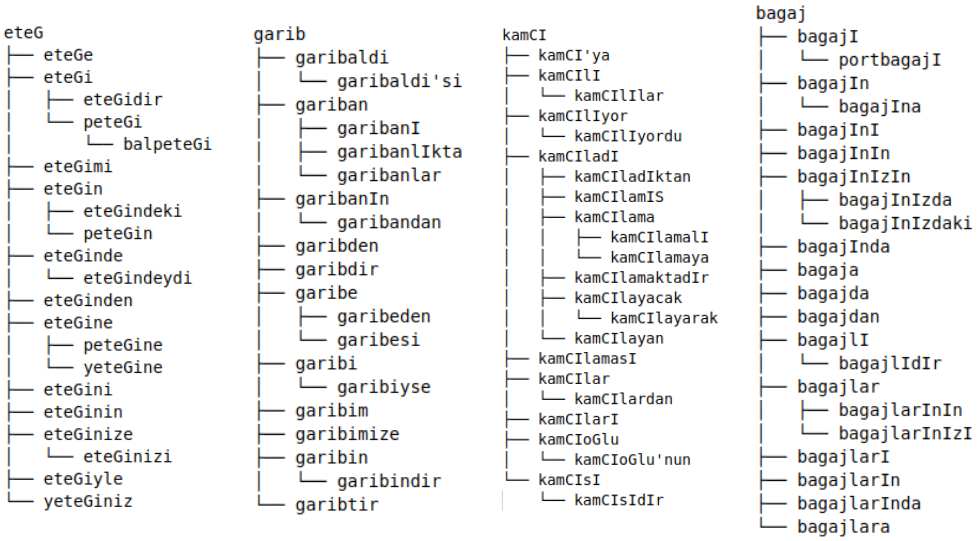


Figure 6. Examples of running our algorithm on 600K Turkish data segmented by supervised and unsupervised Morfessor. In the created trees, nodes are the words and every edge between the nodes represents morphologically relation between words. For example, in the left tree, the root word means "Skirt" and the compounds like "his / her skirt" and "in the his / her skirt" are it's children.

5.2. Error Analysis

In this section, we present an error analysis based on our observations. In the first experiment, when we created a morphological network using the hand-segmented lexicon and the whole procedure was automatic (Section 4.1), we explored two different error types. The first one appeared when we wrongly labeled a root morpheme as an affix (if it was ranked among top-frequent morphemes). For example, as can be seen in Table 1, the word "شَاه [shaah]", which means "king" and ranked 40, is a root morpheme, but we automatically labeled it as a non-root. The second common

type of error happened when our method classified a non-root morpheme as a root morpheme. For example, morpheme "ون [oon] (plural suffix)" was classified wrongly as a root morpheme by our algorithm.

In the second experiment (Section 4.2), we solved the first problem by checking the frequent morphemes manually, and as we expected, the accuracy of the result was better compared with automatic non-root selection. However, the second problem (false roots) still existed. The main reason of this problem is that there are not enough words in our segmented lexicon, and thus our algorithm is not able to identify correct parts of rare words as their root morphemes.

In our third experiment (i.e. expanding the existing graph by adding unseen words), which is described in Section 4.3, the main reason of observed errors was the wrong segmentation for some of the newly added words. It means in some cases, Morfessor offered an incorrect segmentation, which consequently led to wrong morpheme detection and wrong parent-child identification. Table 7 shows five examples of wrong segmentation of supervised and unsupervised Morfessor for our test words. Moreover, in some cases, there was not any child and parent word for test words and consequently, our algorithm could not expand the graph correctly based on them. However, this error happened very few times, while our primary graph was big enough.

The main reason for most faults in our fourth experiment was the wrong segmentation, which is a consequence of having too limited training data. Especially for supervised segmentation of Turkish data, there is simply not enough segmented data available. Also, our observations show that the segmentation boundary count in an average Turkish word is higher than in the case of Persian. Based on this observation we can say Turkish agglutination is more extensive than Persian. Besides, the observations show that the derivational trees for Turkish are in average much deeper than the Persian ones. It also could be a result of higher agglutination.

In Section 4.4, the goal was to compare two methods on two different languages. We hypothesize that the higher segmentation accuracy as well as the higher derivational network accuracy for Persian is a consequence of less extensive agglutination compared to Turkish.

6. Conclusions and future work

In this work, we developed and empirically evaluated an algorithm for creating a morphological (derivational and inflectional) network using a morpheme-segmented lexicon. Our algorithm tries to find all root candidates automatically and creates connections for all words of the lexicon. In addition, we evaluated a modification of our procedure based on a hand-validated set of non-root morphemes.

In the second part of this work, we tried to expand the morphological network by adding 1500 new words into the existing network. While this procedure is automatic, we tried to segment new test words using both supervised and unsupervised versions

word	correct segmentation	unsup. Morfessor	sup. Morfessor
آبزی [aabzi]	آب ـ زی	آبزی	آب ـ ز ـ ی
آبشُشها [aabshoshha]	آب ـ شُش ـ ها	آبشُش ـ ها	آب ـ ش ـ ش ـ ها
تعهدنامه [taahodnameh]	تعهد ـ نامه	ت ـ عهدنامه	ت ـ عهد ـ نامه
بی‌اِجازه [biejaazeh]	بی ـ اِجازه	ب ـ ی ـ اِجازه	ب ـ ی ـ اِجازه
حاکمیت [haakemiat]	حاکم ـ یت	حاکمیت	ح ـ آک ـ میت

Table 7. Sample segmentation of supervised and unsupervised Morfessor for test words in the Persian language.

of Morfessor, an automatic segmentation toolkit. These segmented morphemes are used as the input of our proposed algorithm to find the parents of new words.

We experimented both with Persian and Turkish; the derivational networks for Persian had better final accuracy, which could be a result of lower agglutination compared the Turkish language.

In addition, we evaluated and compared the usage of two unsupervised segmentation algorithms (i.e., Morfessor and MorphSyn) and experimental results showed the better segmentation leads to a more accurate network.

Acknowledgments

The research was supported by OP RDE project No. CZ.02.2.69/0.0/0.0/16_027/0008495, International Mobility of Researchers at Charles University, and by grant No. 19-14534S of the Grant Agency of the Czech Republic. It has been using language resources developed, stored, and distributed by the LINDAT/CLARIAH CZ project (LM2015071, LM2018101). The authors would like to thank all people who listed below who helped us to collect and create the dataset:

- Alireza Abdi
- Sahar Badri
- Abbas Beygi
- Shoeila Behrouznia
- Aysan Chehreh
- Matin Ebrahimkhani
- Aryan Fallah
- Fatemeh Fallah
- Seyed Amirhossein Hosseini
- Amirhossein Mafi
- Zohreh Kazemi
- Nazanin Pakdan
- Seyed Ahmad Sharifi

Bibliography

Ansari, Ebrahim, Zdeněk Žabokrtský, Hamid Haghdoost, and Mahshid Nikravesh. Persian Morphologically Segmented Lexicon 0.5, 2019a. URL https://hdl.handle.net/11234/1-3011. LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Ansari, Ebrahim, Zdeněk Žabokrtský, Mohammad Mahmoudi, Hamid Haghdoost, and Jonáš Vidra. Supervised Morphological Segmentation Using Rich Annotated Lexicon. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019)*, pages 52–61, Varna, Bulgaria, September 2019b. INCOMA Ltd. doi: 10.26615/978-954-452-056-4_007. URL https://www.aclweb.org/anthology/R19-1007.

Arabsorkhi, Mohsen and Mehrnoush Shamsfard. Unsupervised Discovery of Persian Morphemes. In *Proceedings of the Eleventh Conference of the European Chapter of the Association for Computational Linguistics: Posters & Demonstrations*, EACL '06, pages 175–178, Stroudsburg, PA, USA, 2006. Association for Computational Linguistics. doi: 10.3115/1608974.1609002. URL http://dl.acm.org/citation.cfm?id=1608974.1609002.

Baranes, Marion and Benoît Sagot. A Language-independent Approach to Extracting Derivational Relations from an Inflectional Lexicon. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 2793–2799, Reykjavik, Iceland, May 2014. European Language Resources Association (ELRA).

Bijankhan, Mahmood, Javad Sheykhzadegan, Mohammad Bahrani, and Masood Ghayoomi. Lessons from building a Persian written corpus: Peykare. *Language Resources and Evaluation*, 45(2):143–164, 2011.

Cao, Kris and Marek Rei. A Joint Model for Word Embedding and Word Morphology. In *Proceedings of the 1st Workshop on Representation Learning for NLP*, pages 18–26, Berlin, Germany, August 2016. Association for Computational Linguistics. doi: 10.18653/v1/W16-1603. URL https://www.aclweb.org/anthology/W16-1603.

Creutz, Mathias and Krista Lagus. Unsupervised Discovery of Morphemes. In *Proceedings of the ACL-02 Workshop on Morphological and Phonological Learning*, pages 21–30. Association for Computational Linguistics, July 2002. doi: 10.3115/1118647.1118650. URL https://www.aclweb.org/anthology/W02-0603.

Creutz, Mathias, Teemu Hirsimäki, Mikko Kurimo, Antti Puurula, Janne Pylkkönen, Vesa Siivola, Matti Varjokallio, Ebru Arisoy, Murat Saraçlar, and Andreas Stolcke. Morph-based Speech Recognition and Modeling of Out-of-vocabulary Words Across Languages. *ACM Trans. Speech Lang. Process.*, 5(1):3:1–3:29, December 2007. ISSN 1550-4875. doi: 10.1145/1322391.1322394. URL http://doi.acm.org/10.1145/1322391.1322394.

Goldsmith, John. Unsupervised Learning of the Morphology of a Natural Language. *Computational Linguistics*, 27(2):153–198, June 2001. ISSN 0891-2017. doi: 10.1162/089120101750300490. URL https://doi.org/10.1162/089120101750300490.

Grönroos, Stig-Arne, Sami Virpioja, Peter Smit, and Mikko Kurimo. Morfessor FlatCat: An HMM-Based Method for Unsupervised and Semi-Supervised Learning of Morphology. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 1177–1185, Dublin, Ireland, August 2014. Dublin City University and Association for Computational Linguistics. URL https://www.aclweb.org/anthology/C14-1111.

Habash, Nizar and Bonnie Dorr. A categorial variation database for English. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, pages 17–23. Association for Computational Linguistics, 2003. doi: 10.3115/1073445.1073458.

Haghdoost, Hamid, Ebrahim Ansari, Zdeněk Žabokrtskỳ, and Mahshid Nikravesh. Building a Morphological Network for Persian on Top of a Morpheme-Segmented Lexicon. In *Proceedings of the Second International Workshop on Resources and Tools for Derivational Morphology*, pages 91–100, 2019.

Harris, Zellig. From phoneme to morpheme. *Language*, 31:209–221, 1955. doi: 10.1007/978-94-017-6059-1_2.

Hathout, Nabil and Fiammetta Namer. Démonette, a French derivational morpho-semantic network. *Linguistic Issues in Language Technology*, 11(5):125–168, 2014.

Jones, William. *A grammar of the Persian language*, volume 5. John Stockdale, 1807.

Kaleta, Zbigniew. Automatic Pairing of Perfective and Imperfective Verbs in Polish. In *Proceedings of the 8th Language and Technology Conference*, 11 2017.

Karimi, Akbar, Ebrahim Ansari, and Bahram Sadeghi Bigham. Extracting an English-Persian Parallel Corpus from Comparable Corpora. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation, LREC 2018, Miyazaki, Japan, May 7-12, 2018.*, 2018.

Kohonen, Oskar, Sami Virpioja, Laura Leppänen, and Krista Lagus. Semi-supervised extensions to Morfessor baseline. In *Proceedings of the Morpho Challenge 2010 Workshop*, pages 30–34, 2010.

Kőrtvélyessy, Lívia. Cross-linguistic research into derivational networks. In *Proceedings of the Second International Workshop on Resources and Tools for Derivational Morphology*, pages 1–4, 2019.

Kyjánek, Lukáš, Zdeněk Žabokrtskỳ, Magda Ševčíková, and Jonáš Vidra. Universal Derivations Kickoff: A Collection of Harmonized Derivational Resources for Eleven Languages. In *Proceedings of the Second International Workshop on Resources and Tools for Derivational Morphology*, pages 101–110, 2019.

Lango, Mateusz, Magda Ševčíková, and Zdeněk Žabokrtský. Semi-Automatic Construction of Word-Formation Networks (for Polish and Spanish). In *Proceedings of the 11th Language Resources and Evaluation Conference*, Miyazaki, Japan, May 2018. European Language Resource Association. URL `https://www.aclweb.org/anthology/L18-1291`.

Lee, Yoong Keok, Aria Haghighi, and Regina Barzilay. Modeling syntactic context improves morphological segmentation. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning*, pages 1–9. Association for Computational Linguistics, 2011.

Litta, Eleonora, Marco Passarotti, and Chris Culy. Formatio formosa est. Building a Word Formation Lexicon for Latin. In *Proceedings of Third Italian Conference on Computational Linguistics (CLiC-it 2016) & Fifth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2016), Napoli, Italy, December 5-7, 2016.*, 2016. URL `http://ceur-ws.org/Vol-1749/paper32.pdf`.

Narasimhan, Karthik, Regina Barzilay, and Tommi Jaakkola. An unsupervised method for uncovering morphological chains. *Transactions of the Association for Computational Linguistics*, 3:157–167, 2015. doi: 10.1162/tacl_a_00130.

Oflazer, Kemal. Two-level description of Turkish morphology. *Literary and linguistic computing*, 9(2):137–148, 1994. doi: 10.1093/llc/9.2.137.

Piasecki, Maciej, Radoslaw Ramocki, and Marek Maziarz. Recognition of Polish Derivational Relations Based on Supervised Learning Scheme. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC-2012)*, pages 916–922, Istanbul, Turkey, May 2012. European Language Resources Association (ELRA).

Poon, Hoifung, Colin Cherry, and Kristina Toutanova. Unsupervised Morphological Segmentation with Log-linear Models. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, NAACL '09, pages 209–217, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics. ISBN 978-1-932432-41-1. doi: 10.3115/1620754.1620785. URL http://dl.acm.org/citation.cfm?id=1620754.1620785.

Poostchi, Hanieh, Ehsan Zare Borzeshi, and Massimo Piccardi. BiLSTM-CRF for Persian Named-Entity Recognition ArmanPersoNERCorpus: the First Entity-Annotated Persian Dataset. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation, LREC 2018, Miyazaki, Japan, May 7-12, 2018.*, 2018.

Rafea, Ahmed A and Khaled F Shaalan. Lexical analysis of inflected Arabic words using exhaustive search of an augmented transition network. *Software: Practice and Experience*, 23 (6):567–588, 1993. doi: 10.1002/spe.4380230602.

Rasooli, Mohammad Sadegh, Ahmed El Kholy, and Nizar Habash. Orthographic and Morphological Processing for Persian-to-English Statistical Machine Translation. In *Proceedings of the Sixth International Joint Conference on Natural Language Processing*, pages 1047–1051, Nagoya, Japan, October 2013. Asian Federation of Natural Language Processing. URL https://www.aclweb.org/anthology/I13-1144.

Sak, Haşim, Tunga Güngör, and Murat Saraçlar. Turkish language resources: Morphological parser, morphological disambiguator and web corpus. In *International Conference on Natural Language Processing*, pages 417–427. Springer, 2008. doi: 10.1007/978-3-540-85287-2_40.

Ševčíková, Magda and Zdeněk Žabokrtský. Word-Formation Network for Czech. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 1087–1093, Reykjavik, Iceland, May 2014. European Language Resources Association (ELRA).

Shafaei, Elnaz, Diego Frassinelli, Gabriella Lapesa, and Sebastian Padó. DErivCELEX: Development and Evaluation of a German Derivational Morphology Lexicon based on CELEX. In *Proceedings of the DeriMo workshop*, 2017.

Šnajder, Jan. DerivBase.hr: A High-Coverage Derivational Morphology Resource for Croatian. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 3371–3377, Reykjavik, Iceland, May 2014. European Language Resources Association (ELRA).

Šojat, Krešimir, Matea Srebačić, Tin Pavelić, and Marko Tadić. CroDeriV: a new resource for processing Croatian morphology. *Proceedings of the Language Resources and Evaluation-LREC*, 14:3366–3370, 2014.

Taghi-Zadeh, Hossein, Mohammad Hadi Sadreddini, Mohammad Hasan Diyanati, and Amir Hossein Rasekh. A new hybrid stemming method for Persian language. *Digital Scholarship in the Humanities*, 32(1):209–221, 11 2015. ISSN 2055-7671. doi: 10.1093/llc/fqv053. URL https://doi.org/10.1093/llc/fqv053.

Underhill, Robert. *Turkish grammar*. MIT press Cambridge, MA, 1976.

Vilares, Jesús, David Cabrero, and Miguel A. Alonso. Applying Productive Derivational Morphology to Term Indexing of Spanish Texts. In Gelbukh, Alexander, editor, *Computational Linguistics and Intelligent Text Processing*, pages 336–348, Berlin, Heidelberg, 2001. Springer Berlin Heidelberg. ISBN 978-3-540-44686-6. doi: 10.1007/3-540-44686-9_34.

Virpioja, Sami, Ville T. Turunen, Sebastian Spiegler, Oskar Kohonen, and Mikko Kurimo. Empirical Comparison of Evaluation Methods for Unsupervised Learning of Morphology. *TRAITEMENT AUTOMATIQUE DES LANGUES*, 52(2):45–90, 2011. ISSN 1248-9433.

Žabokrtský, Zdeněk, Magda Ševčíková, Milan Straka, Jonáš Vidra, and Adéla Limbursk á. Merging Data Resources for Inflectional and Derivational Morphology in Czech. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, pages 1307–1314, Portorož, Slovenia, May 2016. European Language Resources Association (ELRA). URL https://www.aclweb.org/anthology/L16-1208.

Zeller, Britta, Jan Šnajder, and Sebastian Padó. DErivBase: Inducing and Evaluating a Derivational Morphology Resource for German. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1201–1211, Sofia, Bulgaria, August 2013. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/P13-1118.

Address for correspondence:
Ebrahim Ansari
ansari@iasbs.ac.ir
Malostranské náměstí 25, 118 00 Praha 1, Czech Republic

PBML

The Prague Bulletin of Mathematical Linguistics
NUMBER 115 OCTOBER 2020 129–142

Extending Ptakopět for Machine Translation User Interaction Experiments

Vilém Zouhar, Michal Novák

Charles University, Faculty of Mathematics and Physics, Institute of Formal and Applied Linguistics

Abstract

The problems of outbound translation, machine translation user confidence and user interaction are not yet fully explored. The goal of the online modular system Ptakopět is to provide tools for studying these phenomena. Ptakopět is a proof-of-concept system for examining user interaction with enhanced machine translation. It can be used either for actual translation or running experiments on human annotators. In this article, we aim to describe its main components and to show how to use Ptakopět for further research. We also share tips for running experiments and setting up a similar online annotation environment.

Ptakopět was already used for outbound machine translation experiments, and we cover the results of the latest experiment in a demonstration to show the research potential of this tool. We show quantitatively that even though backward translation improves machine-translation user experience, it mainly increases users' confidence and not the translation quality.

1. Introduction

Internet users often find themselves in need to produce a text in a foreign language they do not speak perfectly. This poses a problem as the users are not able to validate the machine translation result. Our goal is to explore this user–computer interaction and to demonstrate what tools may help the users in these scenarios, increasing their confidence in the produced translations.

For this purpose, we describe Ptakopět, a system which can help with outbound translation. Moreover, it can be extended by other tools and offers an environment for examining usage strategies of outbound translation.

 Corresponding author: zouhar@ufal.mff.cuni.cz
Cite as: Vilém Zouhar, Michal Novák. Extending Ptakopět for Machine Translation User Interaction Experiments. The Prague Bulletin of Mathematical Linguistics No. 115, 2020, pp. 129–142. doi: 10.14712/00326585.008.

1.1. Machine Translation Usage

Machine translation usage can be for some purposes broadly divided into inbound and outbound translation. In inbound translation, mostly gisting, we are the recipients of a message in a foreign–language and it is our responsibility to understand it correctly. It is typically reading websites and e-mails in a foreign language. This use case is characterized by lower quality requirements and the translation not being distributed further.

In outbound translation, the direction of the message is from us to someone else and it is our responsibility to ensure that the message is grammatical- and content-wise correct. An example here is communication by e-mail or filling in foreign language forms. The quality standard here is higher than in gisting.

Reasonable users would not blindly trust the output of a publicly available MT service. Further, they would not paste it into an e-mail and would not send it to someone. In both cases, inbound and outbound translation, feedback on quality is needed. This is true especially in outbound translation because small grammatical errors in inbound translation do not prevent understanding of the message. They, however, do matter in outbound translation because ungrammatical messages could lead to being perceived as unprofessional. This feedback on translation quality should tell users if the translation is correct and if not, which parts contain errors.

The goal is also to increase the users' confidence in machine translation, which, however, cannot be done just by always reporting that everything is correct. To build trust, the whole complex MT service needs to look reliable, that is, to report on adequate occasions that the MT failed and what to do to fix it.

1.2. Existing Approaches

The most rudimentary form of outbound translation solution, especially when the target language is completely unknown to the users, is to perform a manual roundtrip translation (machine translating the result of the forward translation back to the original language). This relies heavily on the assumption that a potential error would only happen in the forward translation and never in the backward translation. This is sometimes not the case. New errors can happen in the backward translation as well and in some cases, the new error may revert the original one. This is shown in the last row (English MT) in Figure 1.

Orthogonal to this is automatic MT quality estimation (QE). The goal of this task is to determine which parts of the forward translation are poorly translated. It is done on word-, phrase-, sentence- or document-level. Companies such as Memsource[1] and Unbabel[2] use QE models (Kepler et al., 2019) to automatically decide which texts

[1] memsource.com/blog/2018/10/01/machine-translation-quality-estimation-memsources-latest-ai-powered-feature/

[2] unbabel.com/blog/unbabel-translation-quality-systems/

svírá úhel	$\xrightarrow{\text{de}}$	Er schließt den Winkel.	$\xrightarrow{\text{cs}}$	Zavírá úhel.
forms an angle		*he closes the angle*		*closes the angle*
svírá úhel	$\xrightarrow{\text{fr}}$	Sait l'angle	$\xrightarrow{\text{cs}}$	Zná úhel
forms an angle		*knows the angle*		*knows the angle*
svírá úhel	$\xrightarrow{\text{en}}$	grips the angle	$\xrightarrow{\text{cs}}$	svírá úhel
forms an angle		*grips the angle*		*forms an angle*

Figure 1. Example of error masking in backward translation in English MT compared to German and French MT in which the forward translation error is revealed. Based on a figure from Zouhar (2020). Capitalization and punctuation preserved from the MT output.

need to be post-edited and to what extent. Such tools are, however, missing in publicly available machine translation services, such as Microsoft Bing Translator, Google Translate or DeepL. The last two services provide alternatives to words and phrases, respectively. Showing alternative translations can also lead to higher user confidence in the system, but it does not help in case the users do not know the target language at all.

1.3. Source Complexity Application

In the context of outbound translation, we would also like to let the users know which parts of the input they should reformulate and focus on to make the output better. Highlighting poorly translated words is the most straightforward application of QE. By using word-alignment between the source text and the translation, we can then estimate which words in the input map to the problematic words in the output.

However, for some MT errors, it is not a specific word that worsens the translation but rather problems in agreements or syntactic structures. Source tokens selected by the described approach are not always responsible for the wrong translation and substituting them may not lead to an improvement. Niehues and Pham (2019) try to model source complexity by comparing the inputs to the training data seen by the MT, which leads to better results than mapping QE to source.

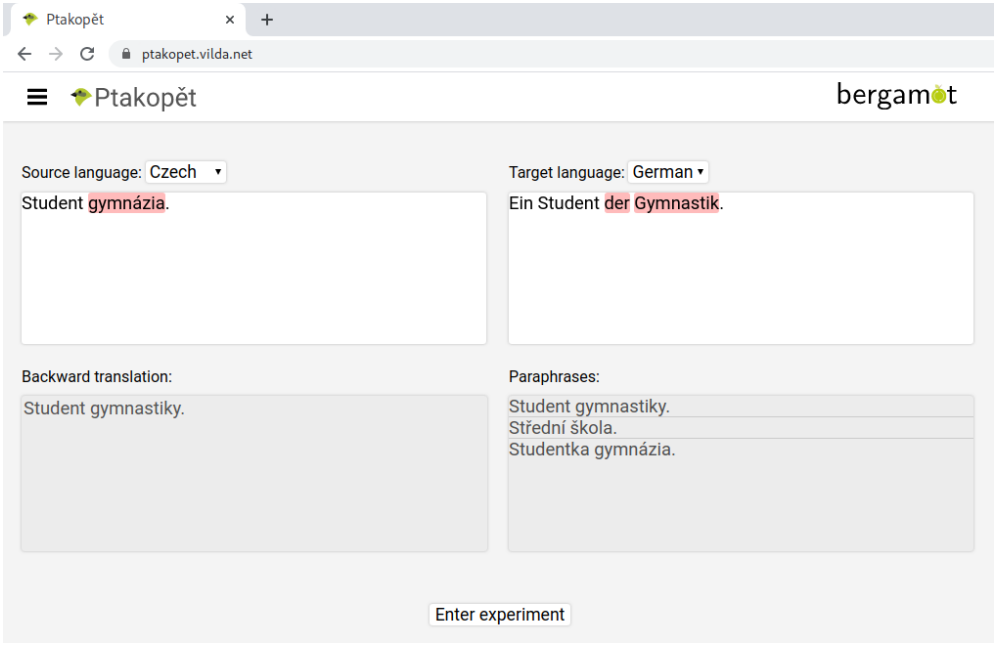


Figure 2. Ptakopět is used to translate a simple Czech noun phrase to German. QE highlights parts of both source and target that were translated incorrectly. Figure from Zouhar (2020).

2. Ptakopět

Ptakopět is a system for outbound machine translation and the exploration of user strategies. It was first presented in Zouhar and Bojar (2020); Zouhar (2020). It is publicly available,[3] the code is open-source[4] and a brief user and technical documentation is also available.[5]

Two other versions of Ptakopět (old-1 and old-2)[6] predate the current version. Their focus was purely outbound translation on the Internet. The current and final version of Ptakopět broadens the focus from only providing an outbound translation tool to also offering a system for analyzing user strategies in dealing with machine translation.

[3] ptakopet.vilda.net

[4] github.com/zouharvi/ptakopet

[5] ptakopet.vilda.net/docs

[6] github.com/zouharvi/ptakopet-old

2.1. Usage

The system offers backward translation, quality estimation, source complexity and paraphrases to help with outbound translation. All of these modules are demonstrated in Figure 2. The top left text area is used for input, the top right for MT output, the bottom left for backward translation and the bottom right for input paraphrases.

The output of the quality estimation is used to highlight erroneous words in the translated output. These words are mapped to the input using word-alignment to estimate problematic source words, which are then also highlighted.

In Figure 2 we see that for the input *student gymnázia* (*grammar school student*) the output *ein student der Gymnastik* (*a student of gymnastics*) appeared instead of *ein Gymnasiast* (*a grammar school student*). Without knowing any German and just by looking at the output, the users could get a false sense of having received a correct translation because the output generally looks like a valid German sentence and the lexemes look similar to what would someone expect in this translation.

Fortunately, this translation error manifested itself in the backward translation and was also detected by quality estimation, which got projected to the source sentence. The affected erroneous parts were highlighted red. The users are now informed that in order to make the translation correct, they must change the last word in the input. For that, the paraphraser module offers several possibilities.[7]

The system is connected to many backends which provide machine translation, quality estimation, word alignment, tokenization, paraphrasing and logging. Naturally, not all backends work with every language pair, and some are more suited for specific testing needs. A menu is shown after clicking the button in the top left corner in Figure 2. It contains settings for switching between the available backends. In Section 3.2 we show how experiment definitions interact with these settings.

2.2. Architecture

The system is composed of three parts: server, frontend and experiment design and data processing suite. The Ptakopět server[8] is used to provide some of the backend services for the frontend, such as quality estimation or tokenization and has no special role compared to other backends except for logs collection.

The frontend is written in TypeScript and is designed to be highly modular. As a result, adding new backend wrappers can be done easily by implementing a single function. Some of these wrappers may not even use the network to resolve requests and compute the result locally as is done, for example, with one tokenization backend wrapper.

[7] Unfortunately none of the three paraphrases suggested in Figure 2 is helpful, each for a different reason.

[8] github.com/zouharvi/ptakopet-server

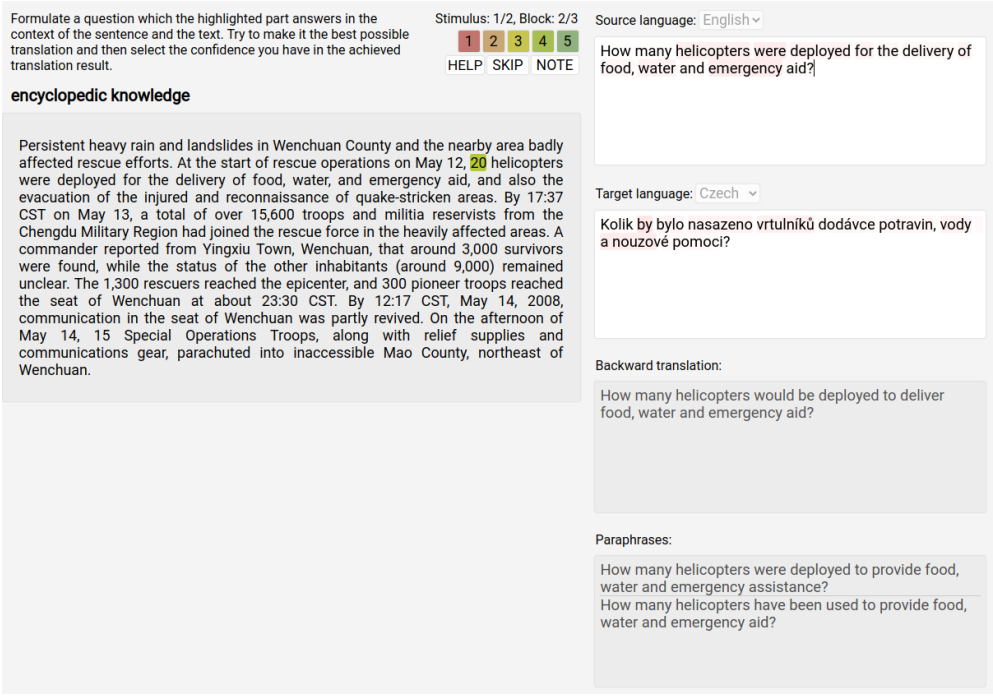


Figure 3. An example stimulus is presented in Ptakopět. Based on this, the users are required to produce a translation in for them an unknown language with the help of offered tools. The target output is content-wise correct, but it is ungrammatical, because the forward MT ommited a preposition "k" ("to" or "for").

3. Deploying and Running Experiments

In this section we demonstrate how experiments in Ptakopět look like and how to design them from the technical point of view.

3.1. Usage

Experiments in Ptakopět are done in the form of showing stimuli to the users and asking them to finish the stimuli with the help of Ptakopět. A stimulus is anything that incentivizes users to produce a text in a foreign language. In Zouhar and Bojar (2020) the stimuli was reporting issues to an IT helpdesk, inquiring into administrative issues and answering encyclopedic questions from the question–answering dataset SQuAD

(Rajpurkar et al., 2018). Technically a stimulus can be any HTML entity, such as text, an image or any more complex web form.

An example stimulus based on SQuAD is shown in Figure 3. A specific piece of information is highlighted to which the users are expected to formulate a question in a foreign language using Ptakopět. In this scenario, we assume that the users speak English and do not speak Czech, so they are not able to evaluate the produced forward translation manually. After they are done working with this current stimulus, they select the level of confidence they have of the produced translation. Multiple events are logged during their work, notably: incoming forward/backward translation, quality estimation, source complexity and paraphrases.

For experiments it is sometimes also desirable to change the configuration settings. This way we can for example enable quality estimation only sometimes (or change the backend) and see whether this change affects the confidence and translation quality.

3.2. Experiment design

An experiment is defined in a single JSON file, which gets loaded when a user tries to log in using *Enter experiment* in Figure 2. Every experiment participant has an assigned user ID (`UID`) by which they log in and are referenced in the experiment definition. We use the concept of *baked queues*. We determine the sequence of stimuli together with their specific configurations for every user in advance as opposed to choosing a random stimulus during the experiment. This way we can check beforehand that the generated queues cover for example every stimulus with a specific number of configurations.

We also present the stimuli in so called *blocks*. They are used only for psychological management purposes so that it is easier for users to split their work into several phases. Users are notified by an alert box every time they completed a block. Our data confirms that there is no connection between the work quality and position of the stimulus in the block.

The experiment definition contains:

- **baked queue:** an array of arrays of stimuli for every user (baked queues in blocks)
- **stimuli dictionary:** a string (valid HTML) for every stimulus identified by stimulus ID (`SID`)
- **configuration rules:** an array of regex rules and changes in settings that get applied when the given stimulus matches the rule's regex

Since the same stimulus can appear in different combinations for different users, the baked queue references stimuli by `SID` Extended (`SIDE`). It is nothing more than a SID with a suffix, separated by a special token `#` (e.g. `p105#bt.y.pp.n.qe.y`). It does get considered when looking up the stimuli content in the stimuli dictionary.

Lastly, the structure contains rules which get applied to the settings whenever the current stimulus matches the regex. For example a rule with the regex `^.*#.*qe\.y.*$`

would match the previous `SIDE` and could then turn the QE on. Multiple settings can be applied at the same time. We find this pattern to be powerful because it allows us to encode the configuration in user baked queues.

4. Results

The pilot experiment in Zouhar and Bojar (2020) suggested that working with an enhanced machine translation system increases production quality. Unfortunately, we then did not collect self-reported user confidence and had every module (except for the paraphraser, which was not part of the experiment) always enabled.

In a small follow-up experiment, we asked 10 annotators to work with Ptakopět on web-form stimuli (e-commerce domain).[9] An excerpt of an online form with a highlighted field was shown (similarly as in Figure 3). The annotators were then asked to fill this field in the target language. We used English for the source language and Czech for the target language. Out of all annotators, 7 knew no Czech, 2 knew Russian (similar to Czech in some aspects) and 1 knew very little Czech. The users were each shown 70 stimuli and they reported their confidence in the produced translation on a scale from 1 (worst) to 5 (best). The 70 stimuli were shared across all users, but one stimulus was seen by different users with different configurations. The configuration was not constant for one user.

We used two MT systems: (1) low–quality, trained on a subsample of 5 million sentence pairs from CzEng 1.7 Bojar et al. (2016), and (2) high–quality, winning MT model of Czech–English News Translation in WMT 2019 (Popel et al., 2019) trained on over 120 million authentic and backtranslated sentence pairs in total.

The enhancement modules comprised backward translation, quality estimation and paraphrases. Backward translation was provided by the abovementioned MT systems trained in the opposite direction. Quality estimation is supplied by a binary supervised classifier, whose word-level predictions are based on glass-box confidence indicators extracted from the output of a neural MT model. Previously, this approach has been successfully employed in sentence-level quality estimation (Fomicheva et al., 2020). We trained the classifier on texts associated with the stimuli collected for this experiment. Paraphrasing was performed using a round-trip translation from English to English through 41 mainly European pivot languages, producing one paraphrase for each pivot language. The MT system used here is based on the Transformer model (Vaswani et al., 2017) sharing the encoder and the decoder for all languages. The annotators were presented only with a selection of paraphrases yielded by 10 higher-resourced pivot languages. The modules were turned on and off during this experiment to see how they affect the confidence and quality.

[9] This experiment is a part of a wider-range experiment in cooperation with University of Edinburgh and University of Sheffield which is still in progress. The complete results of this experiment will be presented in a separate publication.

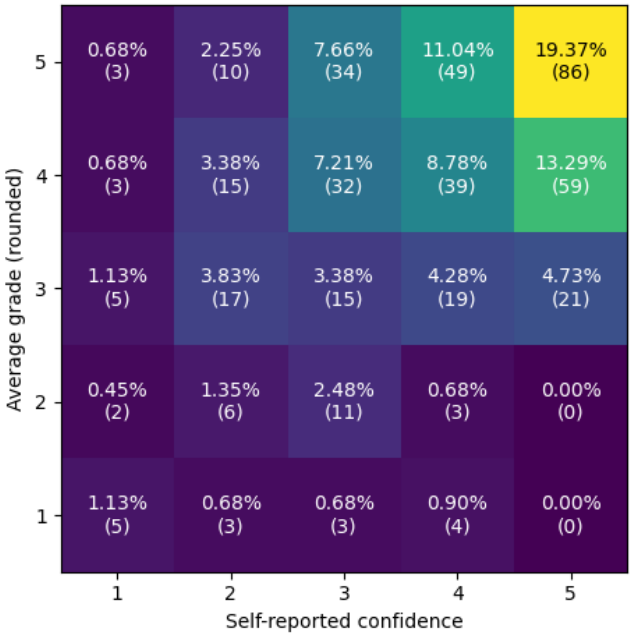


Figure 4. Heatmap of the average rounded grade and self-reported user confidence showing the distribution (all numbers sum to 1). First row in each cells shows the percentage and the second the number of instances. Both axes are 1 (worst) to 5 (best).

The results were then graded by 3 native Czechs on the scale from 1 (worst) to 5 (best). In Figure 4, we show the heatmap for self-reported confidence and translation quality scores. The distribution mass is concentrated in the upper right part of the graph (both high confidence and good translation quality) with very few outliers where the confidence did not match the translation quality grade.

The Fleiss' kappa between the native Czech speakers was 0.36 and the average Pearson correlation coefficient was 0.68. The Pearson correlation coefficient between the average self-reported confidence and average translation quality grade was 0.38.

The relationship between the number of tokens and confidence and quality scores is shown in Figure 5. The translation quality decreases with source sentence length. This is expected because longer sentences are usually more complex and harder to translate. The confidence follows a similar pattern, only with slightly more noise. This same trend could be the result of users correctly identifying errors in long translations using the provided tools. Their judgement could also be based on their apriori knowledge and experience with MT systems which perform poorly on longer sentences.

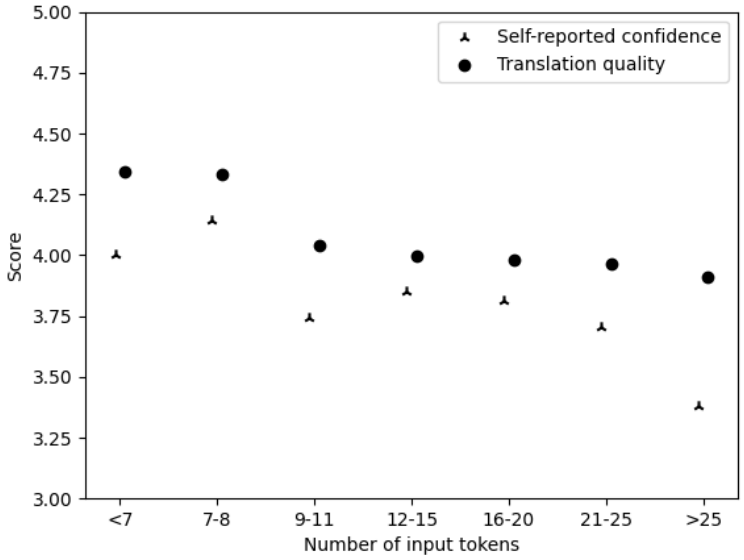


Figure 5. Relationship of the input sentence length and received confidence and translation quality score. Both decline with increased source length.

	Low-quality MT	High-quality MT
Translation quality	3.91	4.18
Self-reported confidence	3.70	3.92
Time-spent	86.95s	77.45s
Interactions	13.17	11.31

Table 1. The effect of MT quality on the average stimulus confidence, translation quality, spent time and the number of user interactions.

Differences in scores, spent time and the number of interactions[10] per MT model are shown in Table 1. With the higher quality of the model, the confidence and translation quality increased by 8.22% and 6.91%, respectively. This means that increasing MT model quality had a higher effect on the translation quality than on the confidence. Users spent on average 9.5 more seconds with the lower–quality MT. This is not because server responses took longer to complete for the lower–quality MT[11] – the number of interactions for the lower–quality MT was also higher. This means that the users used the interface more even though they had not been told what MT model was responding to their translation requests.

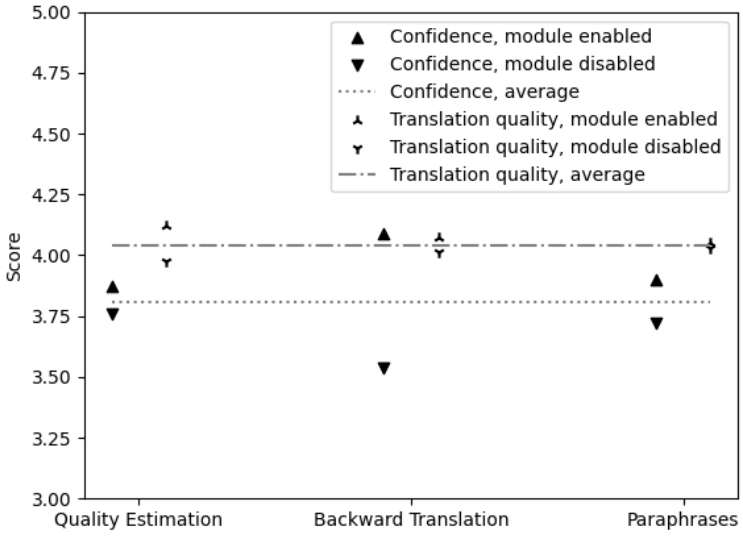


Figure 6. The effect of module presence on confidence and translation quality. Modules help in all cases but to varying degrees.

[10]This was measured by the number of backward translation requests which are started every time forward translation finishes (started upon source input) or the users manually edit the output.

[11]The average translation request duration for an 8-token sentence was 3.2 seconds.

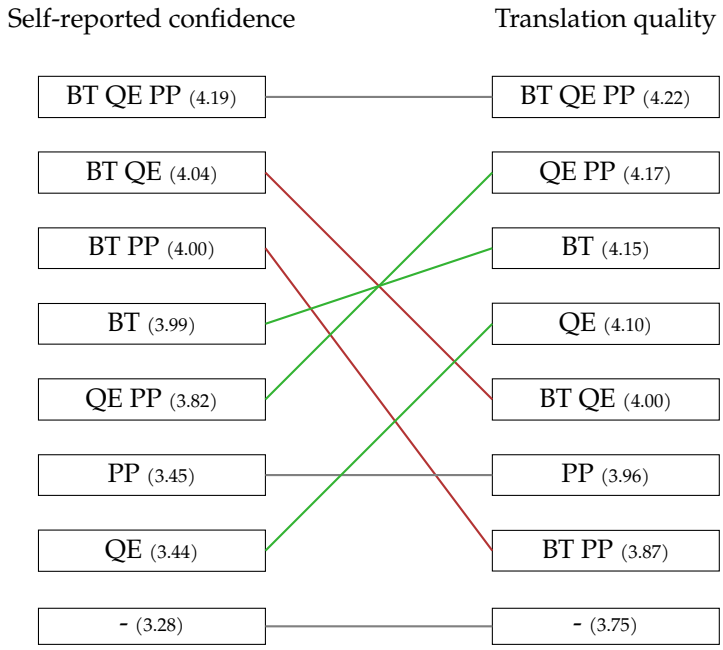


Figure 7. Lists of module configurations sorted by self-reported user confidence (left) and translation quality rated by native Czech speakers (right). BP: backtranslation, QE: quality estimation, PP: paraphrases.

Changing the configuration settings during the experiment helped us in examining which modules helped the most during this task. In Figure 6, we see how the presence or absence of a module affected user confidence and translation quality. In all cases, the presence of a specific module did not worsen the confidence nor the quality. The changes in quality are, however, much less significant than the changes in user confidence. This is most notable in backward translation for which the difference in confidence is 0.55, but the difference in quality is only 0.06.

In Figure 7 we show two columns in comparison. The left column lists configurations sorted by the average self-reported user confidence while the right one lists configurations sorted by the translation quality, respectively. The position of configurations with all modules on (BT QE PP) or off (-) is preserved, but there are many changes in the position of other configurations.

From these figures, we see that any extra module helps in increasing both confidence and translation quality. This refines the previous results that especially backward translation improves machine translation user experience. It does improve it, but it mainly increases users' confidence and not the translation quality.

5. Conclusion

In this article, we described the issue of outbound translation and user confidence in machine translation. We focused on the system Ptakopět and elaborated on the way by which experiments on human annotators are designed in this tool and the design patterns we found useful in the context of online annotation environments.

Finally, in Section 4 we showed some of the results of experiments done using this system. They suggest that enhancements in the form of backward translation, quality estimation and paraphrases help in increasing user confidence more than objective translation quality.

The role of the user is often overlooked in MT research, which is in stark contrast to the fact that there exist tools usable by the users that affect both the confidence and the quality. In future experiments, we would like to extend the functionality of Ptakopět even further to describe the effect of possible enhancement tools for MT rigidly.

Acknowledgements

This project has received funding from the grants H2020-ICT-2018-2-825303 (Bergamot) of the European Union and 19-26934X (NEUREM3) of the Czech Science Foundation.

We also used language resources developed and/or stored and/or distributed by the LINDAT-Clarin and LINDAT/CLARIAH-CZ projects of the Ministry of Education of the Czech Republic (projects no. LM2010013 and LM2018101).

Bibliography

Bojar, Ondřej, Ondřej Dušek, Tom Kocmi, Jindřich Libovický, Michal Novák, Martin Popel, Roman Sudarikov, and Dušan Variš. 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 Artificial Intelligence, pages 231–238, Cham / Heidelberg / New York / Dordrecht / London, 2016. Springer International Publishing. doi: 10.1007/978-3-319-45510-5_27.

Fomicheva, Marina, Shuo Sun, Lisa Yankovskaya, Frédéric Blain, Francisco Guzmán, Mark Fishel, Nikolaos Aletras, Vishrav Chaudhary, and Lucia Specia. Unsupervised Quality Estimation for Neural Machine Translation. *Transactions of the Association for Computational Linguistics*, 8(0):539–555, 2020. doi: 10.1162/tacl_a_00330.

Kepler, Fábio, Jonay Trénous, Marcos Treviso, Miguel Vera, and André F. T. Martins. OpenKiwi: An Open Source Framework for Quality Estimation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics–System Demonstrations*, pages 117–122, Florence, Italy, 2019. Association for Computational Linguistics.

Niehues, Jan and Ngoc-Quan Pham. Modeling Confidence in Sequence-to-Sequence Models. In *Proceedings of the 12th International Conference on Natural Language Generation*, pages

575–583, Tokyo, Japan, 2019. Association for Computational Linguistics. doi: 10.18653/v1/W19-8671.

Popel, Martin, Dominik Machaček, Michal Auersperger, Ondřej Bojar, and Pavel Pecina. English-Czech Systems in WMT19: Document-Level Transformer. In *Proceedings of the Fourth Conference on Machine Translation* (*Volume 2: Shared Task Papers, Day 1*), pages 342–348, Florence, Italy, 2019. Association for Computational Linguistics. doi: 10.18653/v1/W19-5337.

Rajpurkar, Pranav, Robin Jia, and Percy Liang. Know What You Don't Know: Unanswerable Questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics* (*Volume 2: Short Papers*), pages 784–789, Melbourne, Australia, 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-2124.

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is All you Need. In *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc., 2017.

Zouhar, Vilém. Enabling Outbound Machine Translation. Bachelor thesis, Charles University, Faculty of Mathematics and Physics, 2020.

Zouhar, Vilém and Ondřej Bojar. Outbound Translation User Interface Ptakopět: A Pilot Study. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 6969–6977, Marseille, France, 2020. European Language Resources Association.

Address for correspondence:
Vilém Zouhar
zouhar@ufal.mff.cuni.cz
Institute of Formal and Applied Linguistics
Faculty of Mathematics and Physics,
Charles University
Malostranské náměstí 25
118 00 Praha 1, Czech Republic

PBML

The Prague Bulletin of Mathematical Linguistics
NUMBER 115 OCTOBER 2020 143–162

Are Multilingual Neural Machine Translation Models Better at Capturing Linguistic Features?

David Mareček,[a] Hande Celikkanat,[b] Miikka Silfverberg,[b]
Vinit Ravishankar,[c] Jörg Tiedemann[b]

[a] Institute of Formal and Applied Linguistics, Faculty of Mathematics and Physics, Charles University
[b] Department of Digital Humanities, University of Helsinki
[c] Department of Informatics, University of Oslo

Abstract

We investigate the effect of training NMT models on multiple target languages. We hypothesize that the integration of multiple languages and the increase of linguistic diversity will lead to a stronger representation of syntactic and semantic features captured by the model. We test our hypothesis on two different NMT architectures: The widely-used Transformer architecture and the Attention Bridge architecture. We train models on Europarl data and quantify the level of syntactic and semantic information discovered by the models using three different methods: SentEval linguistic probing tasks, an analysis of the attention structures regarding the inherent phrase and dependency information and a structural probe on contextualized word representations. Our results show evidence that with growing number of target languages the Attention Bridge model increasingly picks up certain linguistic properties including some syntactic and semantic aspects of the sentence whereas Transformer models are largely unaffected. The latter also applies to phrase structure and syntactic dependencies that do not seem to be developing in sentence representations when increasing the linguistic diversity in training to translate. This is rather surprising and may hint on the relatively little influence of grammatical structure on language understanding.

1. Introduction

There have been indications that explicitly modeling linguistic information can help performance of neural machine translation (NMT) models (Aharoni and Goldberg, 2017; Nadejde et al., 2017). Conversely, there is evidence that encoder-decoder

 Corresponding author: marecek@ufal.mff.cuni.cz
Cite as: David Mareček, Hande Celikkanat, Miikka Silfverberg, Vinit Ravishankar, Jörg Tiedemann. Are Multilingual Neural Machine Translation Models Better at Capturing Linguistic Features?. The Prague Bulletin of Mathematical Linguistics No. 115, 2020, pp. 143–162. doi: 10.14712/00326585.009.

NMT models also discover linguistic properties without overt supervision while learning to translate (Conneau et al., 2018a; Mareček and Rosa, 2019). This paper provides a new perspective on the topic of linguistic information that is captured by NMT models. Specifically, we investigate the effect of training NMT models on multiple target languages using the assumption that the integration of multiple languages and the increase of linguistic diversity will lead to a stronger representation of syntactic and semantic features captured by the model. Indeed, our experiments show evidence that increasing the number of target languages forces the NMT model to generate more semantically rich representations for input sentences. However, our results do not provide strong support for the integration of additional syntactic properties in latent representations learned by multilingual translation models.

In a bilingual translation setting, especially when the source and target language are related, an NMT model can focus on shallow transformations between the input and output sentences. We hypothesize that this strategy is not sufficient anymore when the number and diversity of the target languages grow. Encoder representations for input sentences in a multilingual setup need to support a mapping to various target language realizations displaying a range of different linguistic properties. In other words, when faced with substantial linguistic diversity, the model will need to create additional abstractions reflecting syntactic and semantic structure that is essential for proper understanding and meaningful translation. In our research, we are interested in finding out what kind of structure is needed in such a setup and what kind of linguistic properties are picked up by current models of attentive neural machine translation.

In order to model a challenging level of linguistic coverage, we, therefore, apply a diverse set of target languages: Czech, Finnish, German, Greek and Italian. Each of these languages exhibit significantly different properties ranging from the complexity of their morphological system and rigidity of word order and syntactic structure up to differences in tense, aspect and lexical meaning. The source language is always English. Based on our experimental setup we now attempt to quantify and compare the semantic and syntactic information discovered by models with increasing amount of target language diversity and we test our hypothesis on two different NMT architectures: The widely-used Transformer architecture (Vaswani et al., 2017), a multi-headed attention based model, and the Attention Bridge architecture (Cífka and Bojar, 2018; Lu et al., 2018), an RNN-based model, which produces fixed-sized cross-lingual sentence representations.

In order to measure linguistic properties discovered by the models, we apply the following three methods: (1) the SentEval linguistic probing tasks on sentence representations, (Conneau et al., 2018a), (2) an analysis of the attention structures regarding the inherent phrase and dependency information (Mareček and Rosa, 2019), and (3) the structural probe on contextualized word representations proposed by (Hewitt and Manning, 2019).

2. Related Work

We learn sentence representations in a multilingual setting. In their seminal paper on multi-lingual neural machine translation, Johnson et al. (2017) show evidence that sentence representations learned for different source languages tend to cluster according to the semantics of the source sentence rather than its language. Schwenk and Douze (2017) train encoder-decoder systems on multiple source and target languages and investigate source sentence representations w.r.t. cross-lingual representation similarity.

Conneau et al. (2018b) train multilingual sentence representations for cross-lingual natural language inference by aligning source and target language representations instead of directly training the system to translate. Artetxe and Schwenk (2019) learn massively multilingual sentence representation on a training set encompassing 93 languages and show good performance on a number of downstream tasks.

Interpretation and evaluation of sentence representations has recently become a very active research area. Conneau et al. (2018a) investigate several ways to learn sentence representations for English and present a benchmark of probing tasks for syntax and semantics.

The structural probe presented by Hewitt and Manning (2019) investigates the relation between the syntax tree of a sentence and its contextualized word embeddings derived from a model. They show that monolingual English ELMo (Peters et al., 2018) and BERT (Devlin et al., 2018) embeddings encode syntactic structure whereas baselines do not. This approach is attractive because it directly investigates syntactic information captured by representations in contrast to probing, where an additional classifier is trained. We apply the structural probe as one of our evaluation methods.

Chrupała and Alishahi (2019) use representational similarity analysis to compare the metrics induced by sentence representations and syntactic dependency trees. This approach is more flexible than the structural probe because it can compare metrics in unrelated spaces (for example continuous sentence representations and symbolic representations like syntax trees).

Another approach to investigate the syntactic information captured by transformer models is to relate self attentions to syntactic phrase or dependency structures. This approach was pioneered by Raganato and Tiedemann (2018), who analyze self attentions in terms of the dependency tree structures and Mareček and Rosa (2019), who train parsers based on self attentions of transformer models in monolingual and multilingual settings.

Whereas there is a large body of related work on interpretation of sentence representations learned by NMT models, few studies directly investigate the effect of multilinguality on sentence representations. Closely related to our work is the work by Ravishankar et al. (2019) which extends the probing tasks presented by Conneau et al. (2018a) into the multilingual domain. They train multilingual sentence representations for NLI by training an English NLI system and mapping sentences from other

languages into the English representation space following Conneau et al. (2018b). They then conduct probing experiments on a multilingual dataset. Ravishankar et al. (2019) notice that, quite surprisingly, transferred representation can deliver better performance on some probing task than the original English representations.

Kudugunta et al. (2019) investigate massively multilingual NMT on a combination of 103 languages. In contrast to this paper, they investigate language representations using Singular Value Canonical Correlation Analysis. They show that encoder representations of different languages cluster according to language family and that the target language affects source language representations in a multilingual setting. In contrast to Kudugunta et al. (2019), our work investigates sentence representations instead of language representations and we investigate the impact of multilinguality on learning syntax and semantics.

To the best of our knowledge, this paper presents the first systematic study of the effect of target language diversity on syntactic and semantic performance for sentence representations learned by multilingual NMT models.

3. Data and Systems

In all our experiments, we use a multi-parallel[1] subset of the Europarl corpus (Koehn, 2005) spanning 391,306 aligned sentences in six languages: English, Czech, Finnish, German, Greek, and Italian. We choose these languages in order to include one representative from each of the major language families in the Europarl dataset allowing maximal diversity among target languages. The multi-parallel corpus is randomly divided into training (389,306 examples), development (1000 examples) and test (1000 examples) sets.

We always use English as the source language, while we vary the number of target languages. Specifically, we set up a systematic study starting with a single target language out of our set, and combining one additional target language at a time, until we reach the exhaustive combination of all the five target languages. Table 1 depicts all our settings. Note that we balance the number of occurrences of each language over training configurations in order to avoid biasing combinations toward particular languages.[2]

We use a multi-parallel corpus in order to avoid injecting additional source language information when increasing the number of target languages. Even when the number of target languages grows, the English source language data remains the same. The only difference is that each source sentence in the training data is paired with multiple translations in each of the target languages. This ensures that any addi-

[1]We took the intersection over the five parallel corpora.

[2]This means that each language occurs twice in 2-combinations, three times in 3-combinations and four times in 4-combinations of languages.

Source	Target	
{En}	**1 tgt**	{Cs}, {De}, {El}, {Fi}, {It}
	2 tgts	{Cs, De}, {De, El}, {El, Fi}, {Fi, It}, {It, Cs}
	3 tgts	{Cs, De, El}, {De, El, Fi}, {El, Fi, It}, {Fi, It, Cs}, {It, Cs, De}
	4 tgts	{Cs, De, El, Fi}, {De, El, Fi, It}, {El, Fi, It, Cs}, {Fi, It, Cs, De}, {It, Cs, De, El}
	5 tgts	{Cs, De, El, Fi, It}

Table 1. The configurations of the 21 different training scenarios. English is the source language in all configurations, while the combination of the target languages differs between scenarios.

tional syntax awareness in models trained on higher combinations of target languages cannot be due to additional English language data.

To preprocess our data, we first run a truecaser (Lita et al., 2003) before splitting into subword units using BPE (Lita et al., 2003). For the latter we train a model with 100k merge operations on the concatenation of all source and target language data.

3.1. Transformer

The first model architecture in our experimental setup is the widely used Transformer model by Vaswani et al. (2017). The Transformer is a multi-headed attention-based, feed-forward architecture. Each head can freely attend to any position, resulting in greater flexibility then competing sequential RNNs. Typically, several layers are stacked on top of each other, and each layer incorporates its own dedicated attention heads. Furthermore, the output from this attention mechanism is averaged with the original input vector via residual connections.

For the Transformer architecture we use a single encoder and decoder even in a multilingual setting using target language labels for informing the translation system about the language to be generated. Following (Artetxe and Schwenk, 2019), we add those labels to the beginning of target sentences rather than source sentences, which effectively hides target language information from the encoder guaranteering a unified source sentence representation. During test time, we force-decode the initial target language label before continuing the standard decoding process that generates the translation in the desired language.

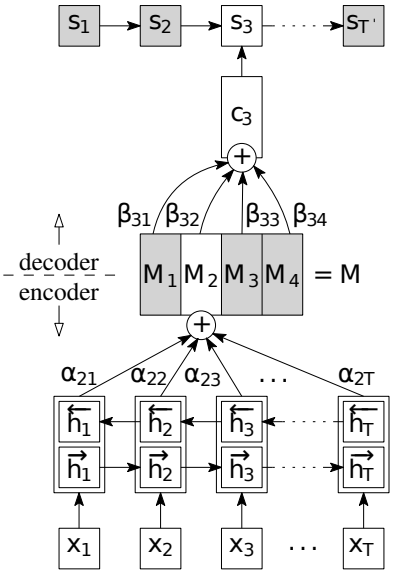


Figure 1. NMT architecture with the attention bridge (Cífka and Bojar, 2018)

3.2. Attention Bridge

Almost all recent NMT architectures (Bahdanau et al., 2015; Vaswani et al., 2017) utilize some kind of cross language attention that directly connects encoder with decoder representations. Cífka and Bojar (2018) introduced the idea of an attention bridge as it is depicted in Figure 1. Here, the whole sentence is encoded into one fixed-size matrix M that serves as an intermediate abstraction layer between attentive encoders and decoders. Sharing this layers across languages enables the effective combination of language-specific encoder and decoder modules to build an extensible multilingual translation architecture. A similar idea was proposed by Lu et al. (2018) but with a slightly different recurrent architecture in the intermediate layer.

In our experiments, we use a variant of the Attention Bridge re-implemented by Raganato et al. (2019) in the OpenNMT-py framework.[3] In this setup we have exactly one encoder for English and one to five separate decoders for our target languages. We run experiments for four different numbers of attention bridge heads: 10, 20, 40, and 80.

[3]Network parameters: 2 bidirectional GRU encoder layers of size 512, MLP attention bridge, 2 GRU decoder layers.

4. Evaluation of Syntax and Semantics

4.1. SentEval Probing Tasks

Our first measure for the degree of semantic and syntactic information captured by sentence representations is a set of ten linguistic classification tasks, so called probing tasks, presented by Conneau et al. (2018a) that look at different syntactic and semantic aspects of a sentence. We conduct experiments using the SentEval toolkit (Conneau and Kiela, 2018) which trains and evaluates models for each of them. Training, development and test data are provided by the SentEval toolkit and we extract the necessary representations for all sentences in those data sets from our Transformer and Attention Bridge models.

Three of the ten SentEval tasks probe for structural properties of the sentence and its syntax tree: **Depth** (depth of the syntax tree), **Length** (binned length of the input sentence) and **TopConsitutents** (the top-most non-root constituents in the syntax tree, for example **NP VP**). Three tasks probe for semantic properties of its main syntactic components: **SubjectNumber** (grammatical number of the subject), **ObjectNumber** (grammatical number of the object) and **Tense** (tense of the main verb). Three of the tasks perturb parts of the original sentences and ask the classifier to identify which of the sentences have been scrambled: **BigramShift** (recognize whether two tokens in the sentence have been transposed), **CoordinationInversion** (recognize whether two coordinated clauses have been transposed) and **SemanticOddManOut** (recognize whether a token in the sentence has been replaced by a random vocabulary item). Finally, **WordContent** is the task of predicting which of around 1,000 mid-frequency words occurs in the input sentence.

WordContent and Length represent surface properties of the sentence; BigramShift, Depth and TopConstituents are purely syntactic tasks; and SubjectNumber, ObjectNumber and Tense are semantic tasks which are related to the syntactic structure of the sentence. Finally, SemanticOddManOut and CoordinationInversion are purely semantic tasks.

We process the training, development and test data for probing tasks identically to the data used for NMT models: we use the same truecasing and BPE models for preprocessing. Subsequently, we extract sentence representations for the sentences to train the SentEval multi-layer perceptron classifier for each task and setting hyperparameters using grid search. Finally, the toolkit provides the classification accuracy on the test set.

4.2. Evaluating Transformer's Self-Attentions

Another way of measuring the amount of syntax captured by the translation encoder is to analyze its self-attention mechanisms and compare them to linguistically motivated syntactic trees (Raganato and Tiedemann, 2018; Mareček and Rosa, 2019).

For this, we partially adapted the approach used by Mareček and Rosa (2019). During the translation of the test data, we extract the weights of the self-attentions of all the attention heads from all six encoder layers, and compare them to syntactic structures of the source sentences automatically created by the Standford Parser (Klein and Manning, 2003) (for phrase-structure trees) and by UDPipe (Straka and Straková, 2017) (for syntactic dependency trees).

An example of typical distributions of weights in one encoder attention head is shown in Figure 2. For our parameter setting,[4] the attentions are very sharp and very often focused on just one token in the previous layer and we observe a kind of continuous phrase attending the same token from the previous layer. Such phrases may then be compared to the syntactic phrases we obtain by a syntactic parser.

The evaluation procedure is the following: First, we "sharpen" the soft attention matrix by only keeping the maximal attention weight on each row of the attention matrix, setting the weights on all other positions to 0:

$$A_{o,i} = \begin{cases} A'_{o,i} & \text{if } A'_{o,i} = \max_{j \in [1,N]} A'_{o,j} \\ 0 & \text{otherwise,} \end{cases} \tag{1}$$

where A' is the original self-attention weight matrix, i and o is the input and output state index respectively, and N is the length of the sentence. Second, we compute the weights for each possible continuous phrase by averaging the individual weights:

$$w_{a,b} = \frac{\sum_{i \in [1,N]} \sum_{o \in [a,b]} A_{o,i}}{b - a + 1}, \tag{2}$$

where a and b is the beginning and the end of the phrase. Such weights are computed for each attention head and for each layer. Then, we can compute layer-wise precision and recall:

$$\mathrm{PhrPrec}_L = \frac{\sum_{h \in H_L} \sum_{[a,b] \in P} w^h_{a,b}}{\sum_{h \in H_L} \sum_{[a,b]} w^h_{a,b}} \tag{3}$$

$$\mathrm{PhrRec}_L = \frac{\sum_{h \in H_L} \sum_{[a,b] \in P} w^h_{a,b}}{|P| \cdot |H|} \tag{4}$$

Where w^h are the phrase weights from attention head h which is chosen from the heads H_L on layer L. P are the phrases present in the constituency tree created by the Stanford Parser.

We can also evaluate the attention matrices with respect to a dependency trees. We simply take the pixels of the attention matrix corresponding to the dependency edges of the dependency tree obtained by UDPipe parser. Since it is not clear whether

[4] layers: 6, heads: 16, ff-size: 4096, normalization: tokens

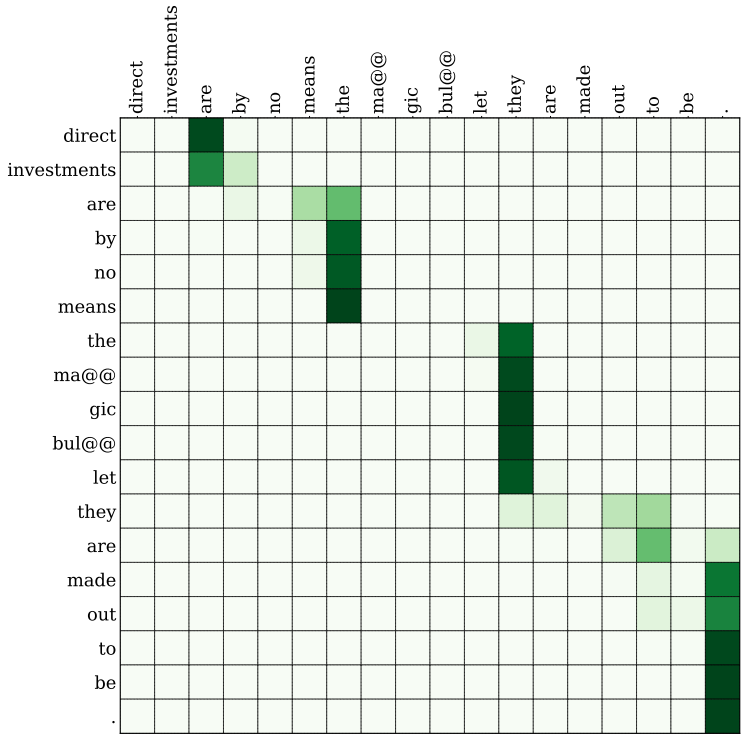


Figure 2. Example of a self-attention head (this one is head 4 on the 3th layer) in transformer encoder. Such continuous phrases attending to the same token are typical for many of the attention heads through all layers.

the dependents should attend to their governors or vice versa, we count both the possibilities. The precision is computed as sum of all "dependency" attention weights divided by the sum of all attention weights.

$$\mathrm{DepPrec}_L = \frac{\sum_{[i,j]\in D} \sum_{h\in H_L} A^h_{i,j} + A^h_{j,i}}{\sum_{h\in H_L} \sum_{i\in[1,N]} \sum_{j\in[1,N]} A^h i,j} \tag{5}$$

The recall is computed as an average weight of "dependency" attention.

$$\mathrm{DepRec}_L = \frac{\sum_{[i,j]\in D} \sum_{h\in H_L} A^h_{i,j} + A^h_{j,i}}{|D| \cdot |H|} \tag{6}$$

4.3. Evaluating Attention Bridge Cross-Attentions

In the attention-bridge architecture, there is one fixed-size vector representation of the input sentence M divided into n vectors composed by the individual attention bridge heads (see Figure 1). Each of them can possibly attend to all sentence tokens but, in practice, they tend to focus on continuous parts of the sentence. An example is included in Figure 3.

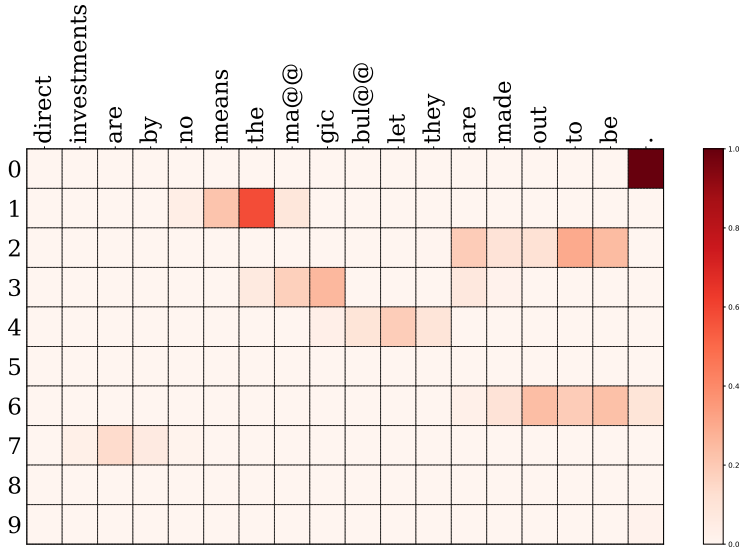


Figure 3. Example of distribution of weights in a 10-headed attention bridge.

Once in a while, we can find more then one phrase per head. However, we treat such cases as one long phrase. For each head we simply take the beginning of the phrase as the leftmost token with weight higher than a threshold t and the end of the phrase as the rightmost token with weight higher than t. We set the threshold t to 0.1. We also tested other thresholds controlling the phrase lengths, but the final results were all very similar and, therefore, we keep the original setting in the results presented hereafter.

Having the set of phrases extracted from the attention bridge, we can now compare it to the phrases of constituency trees obtained by Stanford parser measuring precision in the usual way.

4.4. Structural Probe

We also attempt to evaluate the syntax our representations store by extending Hewitt and Manning's (2019) probe to a multilingual domain. The probe they describe is capable of learning to reliably extract some form of dependency structure, via a combination of two independent distance and depth components. For a detailed mathematical description of either component, we refer the reader to the original paper. Whilst the original probe returns undirected edge weights and depths separately, we (trivially) combine these by forcing edges to point from shallower to deeper nodes. We employ Chu-Liu/Edmonds' algorithm (Chu and Liu, 1965; McDonald et al., 2005) to extract the minimum spanning arborescence of this graph, which is equivalent to a conventional dependency tree.

5. Results

SentEval Probing Tasks: The results of SentEval evaluations are illustrated in Figure 4. For the Attention Bridge, accuracy on all probing tasks except WordContent and SemanticOddManOut generally improves when the number of target languages goes up. The same trend can be seen with all sizes of the attention bridge.

For the Transformer, the effect of adding more target languages does not result in a clear change in probing task accuracy. For Length and Tense, we can discern a small improvement but for the other tasks, performance seems largely independent of the number of target languages. Interesting is that the performance of higher layers is better than for lower layers in almost all cases. SemanticOddManOut is a clear exception. Furthermore, we can also see that the Attention bridge model performs better on most of the probing tasks when adding multiple target languages and increasing the size of the attention bridge. This especially true with the semantic tasks in SentEval.

Syntactic Evaluation of Attentions: Next, we try to assess the attention vectors from the two models in terms of the syntactic information they include. Figure 5 shows the precision and recall results for the phrase trees and the dependency relations. We observe almost no changes or even a slight decreases for the Attention Bridge model when adding more languages to the model. For the Transformer models, we see a slight increase of Phrase precision and recall on the last two layers (4 and 5), whereas the measures on the lower layers are slightly decreasing with the number of target languages.

Structural Probe: Finally, we perform an analysis of the contextualized word representations of the Transformer.[5] Figure 6 describes the variation in UAS with sentence

[5]Note that the Attention Bridge does not produce a per-token representation, and, therefore, this part of the analysis is not applicable for that model.

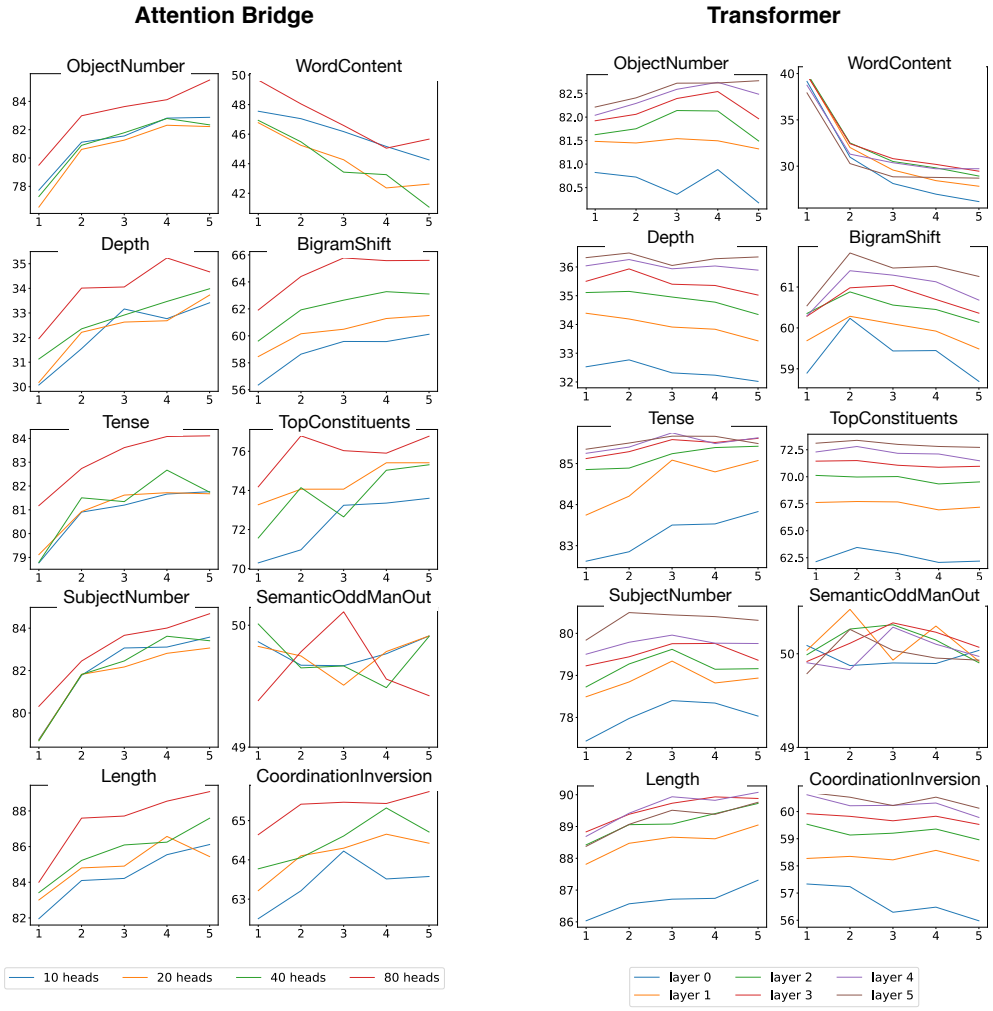


Figure 4. SentEval results for all probing tasks for both the Attention Bridge and Transformer models. The average classification accuracies on the corresponding SentEval task for increasing number of target languages in the models (x-axis) are depicted. For Attention Bridge models, different plot colors indicate different numbers of heads (10, 20, 40, or 80). For Transformer models, different plot colors indicate the layer number (from 0 to 5).

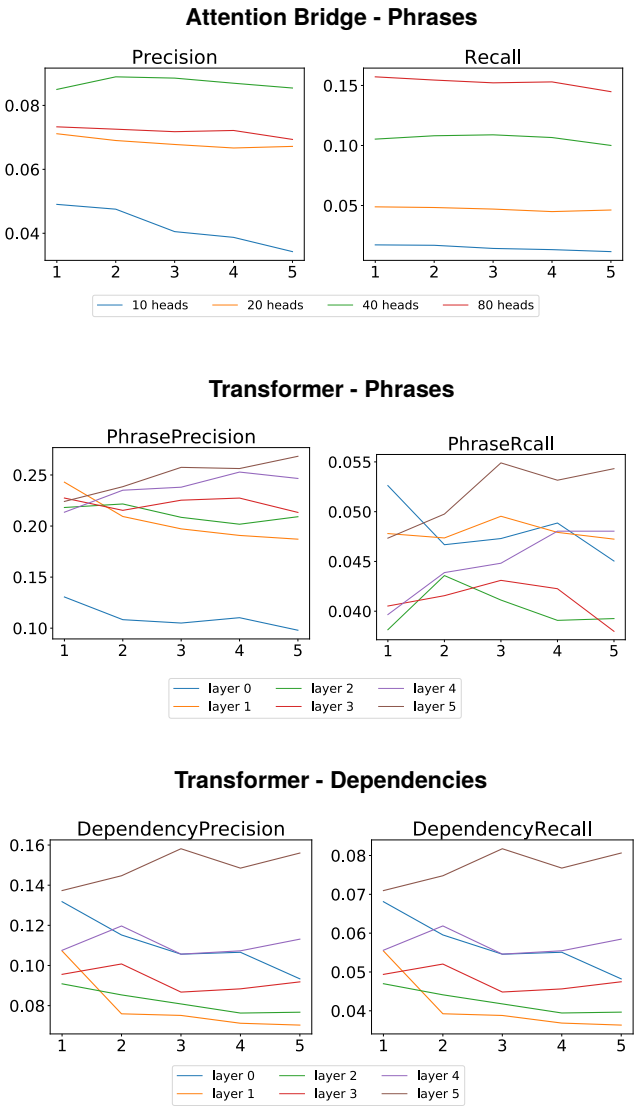


Figure 5. The precision and recall graphs for the continuous phrases extracted from the attention vectors of Attention Bridge and for the continuous phrases and dependency relations form the Transformer models. X-axis denotes the number of target languages.

length for increasing number of languages, and Figure 7 shows UAS variation per token, for three token 'categories' based on their POS.

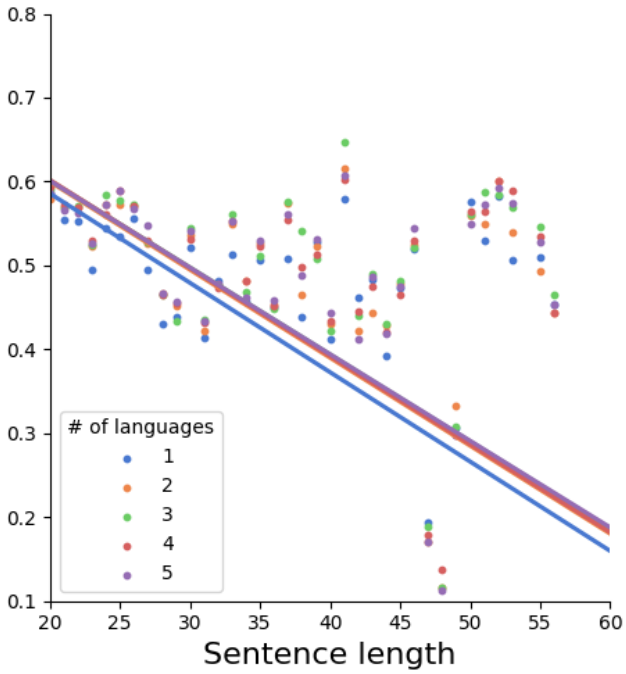


Figure 6. UAS plotted against sentence length. Lines represent trend lines.

6. Discussion

Our results support a connection between the number of target languages in an NMT model and the linguistic properties it picks up at least in the Attention Bridge model as evidenced by the SentEval probing tasks. In that model, all probing tasks except WordContent and SemanticOddManOut significantly increase when the number of target languages in the model grows.

At the same time, BLEU scores for translation performance actually degrade for smaller models (Attention Bridge with 10 and 20 heads) and remains constant for larger models (Attention Bridge with 40 and 80 heads, as well as Transformer), see Figure 8. Degradation of translation performance in itself is not unusual. For example, Kudugunta et al. (2019) notice that performance of high resource languages degrades

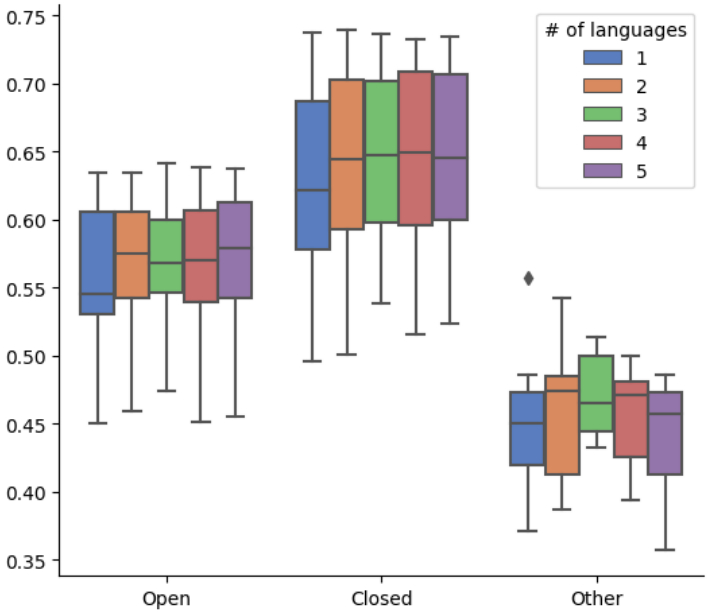


Figure 7. UAS for different groupings of (dependent) tokens by POS. Mappings are the same as in `universaldependencies.org/u/pos/`

in multilingual models. However, it is very interesting that this is accompanied by improved performance on linguistic probing tasks.

For the Transformer model, only the Tense and Length probing tasks seem to show consistent improvement when the number of target languages increases. In general, higher layers tend to deliver a better performance. The overall result for the Transformer model is lower on SentEval tasks than for the Attention Bridge model. This is consistent with some earlier observations, eg., (Tran et al., 2018) who show that RNN-based models tend to outperform the Transformer in subject-verb agreement.

The WordContent task shows a clearly degrading performance when the number of target languages increases. The SemanticOddManOut task in turn shows a very diffuse picture. Those trends are visible in both model architectures, However, these probing tasks differ from all the other ones in the sense that the output label is a word type rather than a category from a limited set or a small integer value as explained in Section 4.1. We believe that the confusion might be due to the BPE segmentation of the input data which generates sub-word level tokens and thus increases the difficulty of

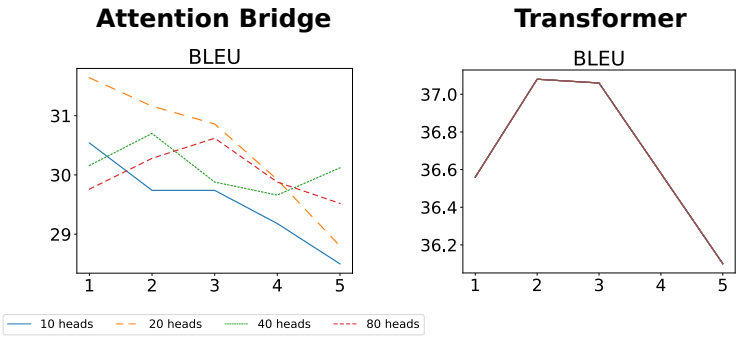


Figure 8. Averaged BLEU scores for Attention Bridge and Transformer. x-axis denotes the number of target languages. Evaluation was done on the test *part of our data.*

the classification task. Furthermore, we note that (Conneau et al., 2018a) also report fluctuating performance for WordContent, which reduces the trust in this particular probing task.

Applying a structural probe to our representations results in several interesting observations. Figure 7 seems to indicate that the jump in *median* syntactic performance is largest when as few as two languages are used as target languages; indicating that the marginal value of further target languages is, as far as syntax is concerned, minimal. Figure 6 also seems to indicate that this holds true across all sentence lengths although the gap widens slightly for longer sentences. We also observe that the increase in median performance is greater for open-class words than closed-class words; this intuitively makes sense, as open-class terms are likelier to have a broader range of semantic values, which are likelier to be better defined with multiple target tokens. Moreover, this observation further corroborates our other results, that exhibit more noticeable improvements in semantic-level tasks than syntactic ones: tokens that receive more reference translations are more likely to be able to better contextualise a broader range of semantic values, particularly from a perspective of lexical disambiguation.

Results for generating syntax trees seem to be largely negative. There is no discernible tendency for the precision or recall on phrase structure for the Attention Bridge model. For the transformer, we see a slight increasing trend in the precision and recall when the number of target languages grows both for phrase structure and dependency parsing for the final layer in the model. There is no clear tendency for the other layers.

An important question our results raise is why the Attention Bridge model shows a much more clear on probing tasks as compared to the Transformer. We hypothesize that this difference may be due to the much greater number of parameters that the

Transformer employs. As a result of having access to a much larger representational space, the Transformer may not have needed to abstract so drastically over several target languages, resorting instead to dedicate some specific part of the representational space to each language. In contrast, the Attention Bridge model with a much more restricted parameter space might have been under more pressure to abstract useful syntactic representations when confronted with a large number of different languages.

7. Conclusion

In this paper, we investigate the impact of additional target languages in multilingual NMT systems on syntactic and semantic information captured by its sentence representations. We analyze two models, the Attention Bridge and the Transformer, using three different evaluation methods. We show evidence that performance on linguistic probing tasks improve for the Attention Bridge when the number of target languages grows. We also show that a transition from a bilingual to a multilingual setting improves performance for the structural probe presented by (Hewitt and Manning, 2019). While we find evidence for improved performance on probing tasks, many of which are related to the semantics of the sentence, our results on syntax performance are inconclusive.

Several interesting unresolved questions remain. Although we tried to cover substantial linguistic variety by using languages from different families, the effect of an even larger typological diversity is still an open question. Additionally, we would also like to know how multiple source languages would affect the results and whether they depend on other latent variables and parameters in the model.

Acknowledgements

This work has been supported by the grant 18-02196S of the Czech Science Foundation and by the FoTran project, funded by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement no 771113). We have been using language resources and tools developed, stored and distributed by theLINDAT/CLARIAH-CZ project of the Ministry of Education, Youth and Sports of the Czech Republic (project LM2018101).

Bibliography

Aharoni, Roee and Yoav Goldberg. Towards String-To-Tree Neural Machine Translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics* (*Volume 2: Short Papers*), pages 132–140, Vancouver, Canada, July 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-2021. URL `https://www.aclweb.org/anthology/P17-2021`.

Artetxe, Mikel and Holger Schwenk. Massively multilingual sentence embeddings for zero-shot cross-lingual transfer and beyond. *Transactions of the Association for Computational Linguistics*, 7:597–610, 2019. doi: 10.1162/tacl_a_00288.

Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio. Neural Machine Translation by Jointly Learning to Align and Translate. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL http://arxiv.org/abs/1409.0473.

Chrupała, Grzegorz and Afra Alishahi. Correlating neural and symbolic representations of language. *arXiv preprint arXiv:1905.06401*, 2019.

Chu, Y. J. and T. H. Liu. On the Shortest Arborescence of a Directed Graph. *Science Sinica*, 14: 1396–1400, 1965.

Cífka, Ondřej and Ondřej Bojar. Are BLEU and Meaning Representation in Opposition? In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1362–1371, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1126. URL https://www.aclweb.org/anthology/P18-1126.

Conneau, Alexis and Douwe Kiela. SentEval: An Evaluation Toolkit for Universal Sentence Representations. *arXiv preprint arXiv:1803.05449*, 2018.

Conneau, Alexis, German Kruszewski, Guillaume Lample, Loïc Barrault, and Marco Baroni. What you can cram into a single vector: Probing sentence embeddings for linguistic properties. *arXiv preprint arXiv:1805.01070*, 2018a. doi: 10.18653/v1/P18-1198.

Conneau, Alexis, Guillaume Lample, Ruty Rinott, Adina Williams, Samuel R Bowman, Holger Schwenk, and Veselin Stoyanov. Xnli: Evaluating cross-lingual sentence representations. *arXiv preprint arXiv:1809.05053*, 2018b. doi: 10.18653/v1/D18-1269.

Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Hewitt, John and Christopher D Manning. A structural probe for finding syntax in word representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4129–4138, 2019.

Johnson, Melvin, Mike Schuster, Quoc Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, Macduff Hughes, and Jeffrey Dean. Google's Multilingual Neural Machine Translation System: Enabling Zero-Shot Translation. *Transactions of the Association for Computational Linguistics*, 5(0), 2017. doi: 10.1162/tacl_a_00065.

Klein, Dan and Christopher D Manning. Fast exact inference with a factored model for natural language parsing. In *Advances in neural information processing systems*, pages 3–10, 2003.

Koehn, Philipp. Europarl: A parallel corpus for statistical machine translation. In *MT summit*, volume 5, pages 79–86. Citeseer, 2005.

Kudugunta, Sneha Reddy, Ankur Bapna, Isaac Caswell, Naveen Arivazhagan, and Orhan Firat. Investigating multilingual nmt representations at scale. *arXiv preprint arXiv:1909.02197*, 2019. doi: 10.18653/v1/D19-1167.

Lita, Lucian Vlad, Abe Ittycheriah, Salim Roukos, and Nanda Kambhatla. Truecasing. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1*, pages 152–159. Association for Computational Linguistics, 2003. doi: 10.3115/1075096.1075116.

Lu, Yichao, Phillip Keung, Faisal Ladhak, Vikas Bhardwaj, Shaonan Zhang, and Jason Sun. A neural interlingua for multilingual machine translation. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 84–92, Brussels, Belgium, October 2018. Association for Computational Linguistics. doi: 10.18653/v1/W18-6309. URL `https://www.aclweb.org/anthology/W18-6309`.

Mareček, David and Rudolf Rosa. From Balustrades to Pierre Vinken: Looking for Syntax in Transformer Self-Attentions. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 263–275, Florence, Italy, August 2019. Association for Computational Linguistics. doi: 10.18653/v1/W19-4827. URL `https://www.aclweb.org/anthology/W19-4827`.

McDonald, Ryan, Fernando Pereira, Kiril Ribarov, and Jan Hajič. Non-Projective Dependency Parsing using Spanning Tree Algorithms. In *HLT-EMNLP*, pages 523–530, 2005. doi: 10.3115/1220575.1220641.

Nadejde, Maria, Siva Reddy, Rico Sennrich, Tomasz Dwojak, Marcin Junczys-Dowmunt, Philipp Koehn, and Alexandra Birch. Predicting target language ccg supertags improves neural machine translation. *arXiv preprint arXiv:1702.01147*, 2017. doi: 10.18653/v1/W17-4707.

Peters, Matthew E, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*, 2018. doi: 10.18653/v1/N18-1202.

Raganato, Alessandro and Jörg Tiedemann. An Analysis of Encoder Representations in Transformer-Based Machine Translation. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 287–297, Brussels, Belgium, November 2018. Association for Computational Linguistics. doi: 10.18653/v1/W18-5431. URL `https://www.aclweb.org/anthology/W18-5431`.

Raganato, Alessandro, Raúl Vázquez, Mathias Creutz, and Jörg Tiedemann. An Evaluation of Language-Agnostic Inner-Attention-Based Representations in Machine Translation. In *Proceedings of the 4th Workshop on Representation Learning for NLP (RepL4NLP-2019)*, pages 27–32, Florence, Italy, August 2019. Association for Computational Linguistics. doi: 10.18653/v1/W19-4304. URL `https://www.aclweb.org/anthology/W19-4304`.

Ravishankar, Vinit, Lilja Øvrelid, and Erik Velldal. Probing Multilingual Sentence Representations With X-Probe. *arXiv preprint arXiv:1906.05061*, 2019. doi: 10.18653/v1/W19-4318.

Schwenk, Holger and Matthijs Douze. Learning Joint Multilingual Sentence Representations with Neural Machine Translation. In *Rep4NLP@ACL*, 2017. doi: 10.18653/v1/W17-2619.

Straka, Milan and Jana Straková. Tokenizing, POS Tagging, Lemmatizing and Parsing UD 2.0 with UDPipe. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw*

Text to Universal Dependencies, pages 88–99, Vancouver, Canada, August 2017. Association for Computational Linguistics. doi: 10.18653/v1/K17-3009. URL http://www.aclweb.org/anthology/K/K17/K17-3009.pdf.

Tran, Ke, Arianna Bisazza, and Christof Monz. The Importance of Being Recurrent for Modeling Hierarchical Structure. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4731–4736, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1503. URL https://www.aclweb.org/anthology/D18-1503.

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is All you Need. In Guyon, I., U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc., 2017. URL http://papers.nips.cc/paper/7181-attention-is-all-you-need.pdf.

Address for correspondence:
David Mareček
marecek@ufal.mff.cuni.cz
Institute of Formal and Applied Linguistics, Faculty of Mathematics and Physics, Charles University, Malostranské náměstí 25, 118 00 Praha, Czechia

PBML

The Prague Bulletin of Mathematical Linguistics
NUMBER 115 OCTOBER 2020 163–186

Derivations and Connections: Word Formation in the LiLa Knowledge Base of Linguistic Resources for Latin

Eleonora Litta, Marco Passarotti, Francesco Mambrini

CIRCSE Research Centre.
Università Cattolica del Sacro Cuore, Milan, Italy

Abstract

The *LiLa* project aims to build a Knowledge Base of linguistic resources for Latin based on the Linked Data framework, with the goal of creating interoperability between them. To this end, LiLa integrates all types of annotation applied to a particular word/text into a common representation where all linguistic information conveyed by a specific linguistic resource becomes accessible. The recent inclusion in the Knowledge Base of information on word formation raised a number of theoretical and practical issues concerning its treatment and representation. This paper discusses such issues, detailing how they are addressed in the project, and introduces the web application to query the collection of lemmas of the Knowledge Base. A number of use-case scenarios that employ the information on word formation made available in the LiLa Knowledge Base are also presented, particularly focusing on the use of the Knowledge Base to compare the perspectives on word formation in different linguistic resources.

1. Introduction

The increasing quantity, complexity and diversity of the currently available linguistic resources for a wide range of languages has led, in recent times, to a growing interest in the sustainability and interoperability of (annotated) corpora, dictionaries,

This paper is an extended version of the work presented by Litta et al. (2019) at the Second Edition of the Workshop on Resources and Tools for Derivational Morphology (DeriMo 2019), 19-20 September 2019, Prague, Czech Republic.

 Corresponding author: marco.passarotti@unicatt.it
Cite as: Eleonora Litta, Marco Passarotti, Francesco Mambrini. Derivations and Connections: Word Formation in the LiLa Knowledge Base of Linguistic Resources for Latin. The Prague Bulletin of Mathematical Linguistics No. 115, 2020, pp. 163–186. doi: 10.14712/00326585.010.

thesauri, lexica and Natural Language Processing (NLP) tools (Ide and Pustejovsky, 2010). This effort, initially, resulted in the creation of databases and infrastructures hosting linguistic resources, such as CLARIN,[1] DARIAH,[2] META-SHARE[3] and EAGLE.[4] Such initiatives collect resources and tools, which can be used and queried from a single web portal, but they do not provide real interconnection between them. In fact, in order to make linguistic resources interoperable, all types of annotations applied to a particular word/text should be integrated into a common representation that enables access to the linguistic information conveyed in a linguistic resource or produced by an NLP tool (Chiarcos, 2012, p. 162).

To meet the need of interoperability, the *LiLa* project's objective (2018-2023)[5] is to create a Knowledge Base of linguistic resources for Latin based on the Linked Data framework,[6] i.e. a collection of multifarious, interlinked data sets described with the same vocabulary of knowledge description (by using common data categories and ontologies). The ultimate goal of the project is to exploit to the fullest the wealth of linguistic resources and NLP tools for Latin developed so far, and to bridge the gap between raw language data, NLP and knowledge description (Declerck et al., 2012, p. 111).

In its design, the structure of LiLa is highly lexically-based: the core component of the Knowledge Base is an extensive list of Latin lemmas extracted from the morphological analyser for Latin Lemlat (Passarotti et al., 2017). This list has been compiled into a database from three reference dictionaries for Classical Latin ((Georges, 1913); (Glare, 1982); (Gradenwitz, 1904)), the entire Onomasticon from Forcellini's (Forcellini, 1867) Lexicon Totius Latinitatis (Budassi and Passarotti, 2016) and the Medieval Latin Glossarium Mediae et Infimae Latinitatis by du Cange et al. (1883-1887), for a total of over 150,000 lemmas (Cecchini et al., 2018). The portion of the lexical basis of Lemlat concerning Classical and Late Latin (43,432 lemmas) was also enhanced with information taken from the Word Formation Latin (WFL) lexicon (Litta and Passarotti, 2019), a lexical resource that provides information about derivational morphology by connecting lemmas via word formation rules.

The consolidation of information taken from WFL into the LiLa Knowledge Base raises a number of theoretical and practical issues concerning the treatment and representation of word formation in LiLa. The present paper discusses such issues, presenting how they are addressed in the project. The paper is organised as follows. Section 2

[1] http://www.clarin.eu.

[2] http://www.dariah.eu.

[3] http://www.meta-share.org.

[4] http://www.eagle-network.eu.

[5] https://lila-erc.eu

[6] See Tim Berners-Lee's note at https://www.w3.org/DesignIssues/LinkedData.html.

introduces the LiLa Knowledge Base, sketching its fundamental architecture. Section 3 presents the WFL lexicon. Section 4 discusses how word formation is accounted for in LiLa, detailing the classes of the LiLa ontology concerned. Section 5 describes the main features of the web application built to query the collection of lemmas of the Knowledge Base. Section 6 presents a number of use-case scenarios that employ the information on word formation made available in LiLa, particularly focusing on the use of the Knowledge Base to compare the perspectives on word formation provided by different linguistic resources. Lastly, Section 7 concludes the paper.

2. The LiLa Knowledge Base

In order to achieve interoperability between distributed resources and tools, LiLa adopts a set of Semantic Web and Linked Data standards. These include ontologies that describe linguistic annotation (OLiA, Chiarcos and Sukhareva, 2015), corpus annotation (NLP Interchange Format (NIF), Hellmann et al., 2013; CoNLL-RDF, Chiarcos and Fäth, 2017) and lexical resources (Lemon, Buitelaar et al., 2011; Ontolex, McCrae et al., 2017[7]). Furthermore, following Bird and Liberman (2001), the Resource Description Framework (RDF) (Lassila et al., 1998) is used to encode graph-based data structures to represent linguistic annotations in terms of triples: (1) a predicate-property (a relation; in graph terms: a labeled edge) that connects (2) a subject (a resource; in graph terms: a labeled node) with (3) its object (another resource, or a value, e.g. a string or an integer). The SPARQL Protocol and RDF Query Language (SPARQL) is used to query the data recorded in the form of RDF triples in a triplestore (Prud'Hommeaux et al., 2008).[8]

The lexically-based nature of the LiLa Knowledge Base results from a simple, fundamental assumption: textual resources are made of (occurrences of) words, lexical resources describe properties of words, and NLP tools process words. In particular, the lemma is considered the ideal interconnection between lexical resources (such as dictionaries, thesauri and lexica), annotated corpora and NLP tools that lemmatise their input text. Lemmas are canonical forms of words that are used by dictionaries to cite lexical entries, and are produced by lemmatisers to analyse tokens in corpora. For this reason, as was said, the core of the LiLa Knowledge Base is represented by the collection of Latin lemmas taken from the morphological analyser Lemlat;[9] Lemlat has proven to cover more than 98% of the textual occurrences of the word forms recorded in the comprehensive *Thesaurus formarum totius latinitatis* (TFTL, Tombeur, 1998), which is based on a corpus of texts ranging from the beginnings of Latin literature up to present times, for a total of more than 60 million words (Cecchini et al., 2018). LiLa thus aims to achieve interoperability by linking all entries in lexical re-

[7] `https://www.w3.org/community/ontolex`.

[8] A prototype of the LiLa triplestore is accessible at `https://lila-erc.eu/sparql`.

[9] `https://github.com/CIRCSE/LEMLAT3`.

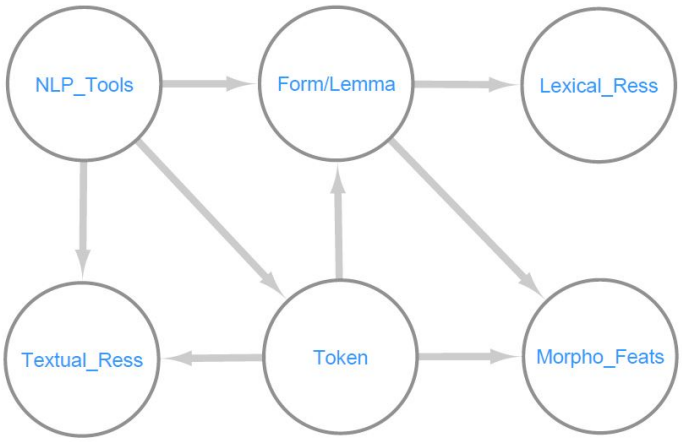


Figure 1. The fundamental architecture of LiLa.

sources and corpus tokens that refer to the same lemma, allowing a good balance between feasibility and granularity.

Figure 1 shows a simplified representation of the fundamental architecture of LiLa, highlighting the relations between the main components represented by the lemma and the other types of resources that interact with the Knowledge Base. There are two nodes representing as many kinds of linguistic resources linked to the core components: a) **Textual Resources**: they provide texts, which are made of **Tokens**; from a morphological standpoint, tokens can be analysed as occurrences of word forms;[10] b) **Lexical Resources**: they describe lexical items, which can include references to lemmas (e.g. in a bilingual dictionary), or to word forms (e.g. in a collection of forms like the aforementioned TFTL). A **Lemma** is one special type of (inflected) **Form** that is conventionally chosen as the citation form for a lexical item. Both tokens and forms (and thus lemmas, as a subclass of forms) are assigned **Morphological Features**, like part of speech (PoS), inflexional category and gender. Finally, **NLP tools** such as tokenisers, PoS taggers and morphological analysers can process respectively textual resources, tokens and forms.

Using the Lemma node as a pivot, it is thus possible to connect resources and make them interact, for instance by searching in different corpora all the occurrences of a lemma featuring some specific lexical properties (provided by one or more lexical resource).

[10]The degree of overlapping between tokens and forms depend on the criteria for tokenisation applied. Given the morphosyntactic properties of Latin, in LiLa this overlapping is complete.

3. The Word Formation Latin Lexicon

The WFL lexicon adds a layer of information on word formation to the lexical materials for Classical and Late Latin of the Lemlat database. The lexicon is based on a set of word formation rules (WFRs) represented as directed one-to-many input-output relations between lemmas. The lexicon was devised according to the Item-and-Arrangement (I&A) model of morphological description (Hockett, 1954): lemmas are either non-derived lexical morphemes, or a concatenation of a base in combination with affixes. This theoretical model was chosen because it emphasises the semantic significance of affixal elements, and because it had been previously adopted by other resources treating derivation, such as the morphological dictionaries Word Manager (Domenig and ten Hacken, 1992).

WFL is characterised by a step-by-step morphotactic approach: each word formation process is treated individually as the application of one single rule. For instance, the adjective *febricula* 'a slight fever' is recorded in WFL as derived from the noun *febris* 'fever' via a WFR that creates diminutive nouns with the suffix -(*us*/*un*)*cul*.

This approach results in a hierarchical structure, whereby one or more lemmas derive from one ancestor lemma. A set of lemmas derived from one common ancestor is defined as a "word formation family". In the web application for querying the WFL lexicon, this hierarchical structure is represented in a directed graph resembling a tree.[11] In the graph of a word formation family, nodes are occupied by lemmas, and edges are labelled with a description of the WFR used to derive the output lemma from the input one. For instance, Figure 2 shows the derivation graph for the word formation family whose ancestor (or "root") lemma is *febris*.

Each output lemma can only have one input lemma, unless the output lemma qualifies as a compound, as in the case of *febrifugia* 'a plant called centaury', a compound formed by the noun *febris* and the verb *fugo* 'to cause to flee, to drive away'. In WFL, simple conversion (i.e. change of PoS without further affixation) is treated as a separate WFR, like in the case of the the verb *fugo* derived from the noun *fuga* 'flight' in Figure 2. However, when considering formations involving both the attachment of an affix and a shift in PoS (as, for example, *febris*, noun >*febricito* 'to have a fever', verb), these are handled in one single step.

That being said, portraying word formation processes via directed graphs raises some significant theoretical issues, especially in cases where the derivational direction is uncertain or unsuitable to be represented by a single step-by-step process (Budassi and Litta, 2017). In such instances, WFL adheres to a strict methodology in order to work around fuzziness. An illustrative case in point is the difficulty in firmly establishing a direction in the derivation of conversion processes such as N-to-A or A-to-N. When considering, to give an example, the relation between the adjective *adversus* 'facing towards', the noun *adversarius* 'an opponent', and the adjective *adversarius* 'hostile',

[11] `http://wfl.marginalia.it`.

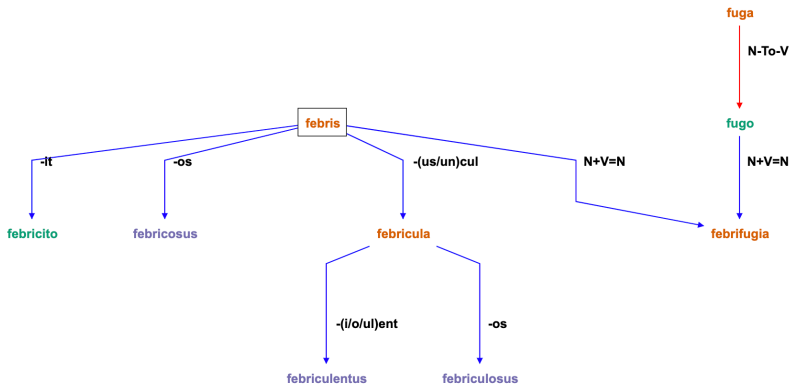


Figure 2. Derivation graph for the word formation family of febris *in WFL.*

did the word formation process work like *adversus* > *adversarius* A > *adversarius* N, or like *adversus* > *adversarius* N > *adversarius* A? When there is space for interpretation on which direction the change has happened, Oxford Latin Dictionary (OLD) (Glare, 1982) is used, as a rule, to "testify" the provenance of lemmas (in our case *adversus* > *adversarius* A > *adversarius* N). Even so, in a few occasions it has been necessary to take some independent choices: for instance, OLD states that diminutive noun *amiculus* 'a pet friend' derives from the adjective *amicus* 'friend'; we, however, chose to make it derive from noun *amicus* as it seems more probable that a diminutive noun was created to diminish a noun rather than an adjective.

The most controversial strategy adopted in WFL to work around non-linear derivations was the creation of "fictional" lemmas that act as placeholders between attested words in order to justify extra "mechanical" (morphotactic) steps. The verb *exaquesco* 'to become water', for example, is connected to the noun *aqua* 'water', through a made-up verb **aquesco*.[12] However, the existence of these fictional lemmas has proven to be less than ideal. User feedback has reported confusion and puzzlement at the presence of the fictional element in the derivational tree. Moreover, when browsing the data, the existence of fictional lemmas needs to be factored in. For instance, if looking for all lemmas created with the suffix *-bil* in WFL, 598 lemmas are given as a result.[13] In WFL, 103 of these are fictional lemmas, 17% of the total number of lemmas derived

[12]The asterisk used to indicate fictional lemmas in WFL does not have the same value as the asterisk employed in Indo-European studies to indicate a reconstructed word, but merely marks a fabricated "stepping stone" in a two-step derivational process.

[13]These are in Latin adjectives that have generally instrumental (e.g. *terribilis* 'by whom/which one is terrified') and/or passive and potential meaning (e.g. *amabilis* 'which/who can be loved') (Kircher-Durand, 1991 and Litta, 2019).

using the *-bil* suffix. The vast majority of these were fabricated in order to establish a derivational process between lemmas such as the adverb *imperabiliter* 'authoritatively' to their "next of kin", the verb *impero* 'to demand, to order'. In order to account for these two steps, i.e. the addition of the suffixes *-bil* and *-ter*, the fictional adjective **imperabilis* was created as a further step in the word formation process. The presence of fictional lemmas in the WFL dataset means that when making general considerations on the distribution of the *-bil* suffix in Classical and Late Latin, for instance, one should keep in mind that a portion of what is extracted from WFL might need to be disregarded.

4. Word Formation in *LiLa*

The inclusion of the WFL data into the LiLa Knowledge Base provided an opportunity to devise a different way to account for those processes that do not fit into a linear hierarchical structure. The recent emergence of interest in the application of Word and Paradigm (W&P) models to derivational morphology (Blevins, 2016) and, in particular, the theoretical framework of the word-(and sign)-based model known as Construction Morphology (CxM) (Booij, 2010), has been crucial for designing the inclusion of the WFL data into LiLa.[14] CxM revolves around the central notion of "constructions", conventionalised pairings of form and meaning (Booij, 2010, p. 6). For example, the English noun *driver* is analysed in its internal structure as $[[\texttt{drive}]_{V}\ \texttt{er}]_{N} \longleftrightarrow [\texttt{someone who drive(s)}_{V}]_{N}$. Constructions may be hierarchically organised and abstracted into "schemas". The following schema, for instance, describes a generalisation of the construction of all words displaying the same morphological structure as *driver*, like for instance *buyer*, *player* and *reader*: $[[\texttt{x}]_{Vi}\texttt{er}]_{Nj} \longleftrightarrow [\texttt{someone who SEM}_{Vi}]_{Nj}$.[15]

One of the most crucial fundamentals of CxM is that schemas are word-based and declarative, which means that they describe static generalisations, as opposed to explaining the procedure of change from one PoS to another like WFRs do (e.g. V-to-N *-er*). Also, schemas are purely output-oriented, so the focus is not on the derivational process anymore, but on the morphological structure of the word itself. This translates into a concept that is especially fit to be included in the LiLa Knowledge Base: if words can be described as a construction of formative elements, these can be organised into (connected) classes of objects in an ontology.

In particular, the LiLa ontology defines three classes of objects that are used for the treatment of derivational morphology: (1) Lemmas, (2) Affixes, divided into Prefixes and Suffixes, and (3) Bases. Each Affix is labelled with a citation form chosen to repre-

[14]For a full description of the theoretical justification of why W&P approaches such as CxM can be advantageous in describing word formation in Latin, see Litta and Budassi (Forthcoming).

[15]Subscript like *V*, *N*, *i* and *j* are traditionally used as placeholders for morphological (e.g. *V* and *N*) and semantic (e.g. *i* and *j*) features that are referred to separately.

sent it in the Knowledge Base. Bases are currently not assigned a further description, nor are they associated with any human-readable string (such as, for instance, a lexical stem); they are simply defined by their function of connectors between lemmas belonging to the same word formation family.[16] Like any object in LiLa, Affixes and Bases are assigned a unique identifier.

These three classes of objects are connected to each other via object properties that are also formalised in the ontology of LiLa. A Lemma node is linked (a) to the Affix nodes that are part of its construction through the relationship `hasPrefix` or `hasSuffix` and (b) to its Base (or Bases, in the case of compounds) through the relationship `hasBase`. No relation of derivational nature is posed between lemmas, so as not to take assumptions on the direction of the formative process.

Figure 3. The word formation family of febris *in LiLa.*

[16] In what follows, therefore, bases will be mentioned using the numeric ID that forms the last component of their URI (e.g. Base 217 is the base that has the URI: `http://lila-erc.eu/data/id/base/217`); occasionally, if no ambiguity arises, they are also cited by mentioning one of the lemmas that are attached to it (thus, the same base can be referred to as "the base of *gigno*").

Figure 3 shows the word formation family of *febris* as it is represented in LiLa. Nodes for Lemma objects are assigned a unique identifier, that can be read by hovering on the node. By expanding the toggles on each node it is possible to view all the information linked to it. For example, the lemma with ID 103049[17] has written representation 'febriculentus', this information can be found on the node and on a list that opens on the right hand side of the screen detailing the type of node, its label and written representation(s). Node 'febriculentus' (which belongs to the class Lemma) is connected through a `hasPOS` property to the node 'adjective', through `hasInflectionType` to 'first class adjective', through `hasDegree` to 'positive', to two suffixes with written representation '-(us/un)cul' and '-(i/o/ul)ent' respectively, and to Base 1633. This base node has 7 ingoing edges, one for each of the lemmas belonging to the word formation family *febris* belongs to. One of these lemmas, *febrifugia* is also related to another base (1719), which connects all members of the word formation family of *fugo*, because it is a compound.

5. Querying the Lemma Collection of LiLa

LiLa provides also a user-friendly interface to query its collection of lemmas.[18] The query results are shown as lists of lemmas, and all the information linked to a lemma in the Knowledge Base can be visualised via a simple LodLive application.[19]

In this section, we describe the query interface of the lemma collection, specifically focusing on the retrieval and visualisation of information about derivational morphology.

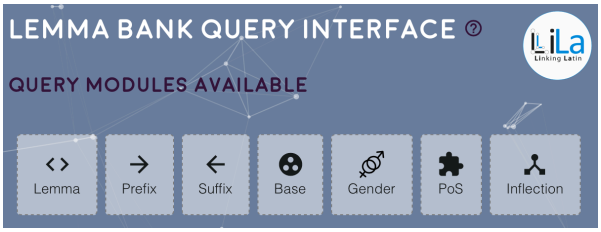


Figure 4. The query interface of the LiLa lemma collection.

Figure 4 shows a screenshot from the query interface home page. Users can select one or more query modules. On selection, the "Lemma" module allows free text (and

[17] `http://lila-erc.eu/data/id/lemma/103049`.

[18] `https://lila-erc.eu/query`.

[19] `http://en.lodlive.it`.

RegEx) input, while the others offer a range of values available from a drop-down menu. The list of lemmas resulting from the combination of values from the modules selected is updated dynamically. For instance, the module-value couples Prefix=*ante-* and PoS=*Verb* return a list of all verbs in the lemma collection that are formed with the prefix *ante-*.

For every query, it is possible to download the results as a CSV file (Comma-Separated Values) and also to copy the SPARQL code for the query, which can be reused (and obviously modified *ad libitum*) on the endpoint of the LiLa Triplestore.[20]

For example, the selection of the Lemma query module with free text value *febris* returns two records with written representation *febris* in the lemma collection, a common noun and a proper noun (both feminine of the third declension). From the resulting list it is possible to consult data on a chosen lemma from two different points of view: a data sheet and a graph view in LodLive.

The data sheet includes the URL for the relevant lemma, its label and written representation(s), its type, links to bases, affixes and suffixes (if any), gender, inflectional category and PoS. Figure 5 shows the data sheet for the common noun *febris*. All information on the data sheet is clickable and brings to other relevant data sheets. For example, the URL and number of the base leads to the dedicated web page of the data point.

The second, more dynamic way of visualising the data is the graph view. Here, lemmas, affixes, bases and other objects from the ontology are shown as circle shaped nodes surrounded by a number of smaller satellite circles. Clicking on each of them reveals all the information linked to the nodes in the Knowledge Base. Figure 6 shows the information linked to base node labelled Base 1633.[21] Beside the entity type of the node (Base), the lemmas that belong to the same family of *febris* are shown. All lemmas are linked to the base node via the `hasBase` property.

The LiLa interface also allows users to run complex queries on derivational information, by combining different modules. Figure 7 shows an example of a query that searches for verbs formed with prefix *de-*, suffix *-sc* and at least one written representation of their cita-

Figure 5. The data sheet of lemma febris.

[20] `https://lila-erc.eu/sparql`.

[21] `http://lila-erc.eu/data/id/base/1633`.

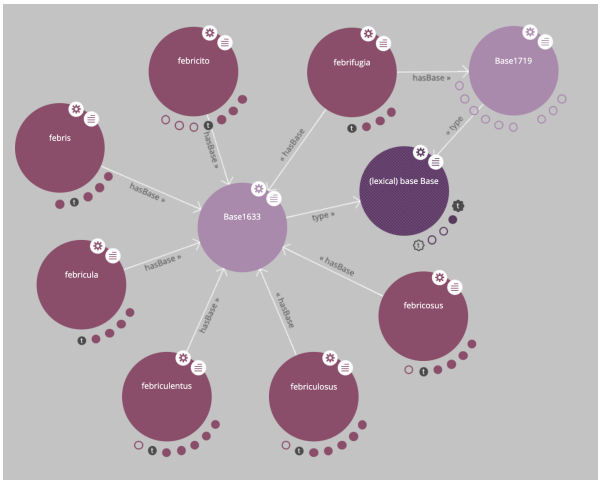


Figure 6. The graph view of the word formation family of febris *in the query interface of the LiLa lemma collection.*

tion form beginning with the characters *de* (expressed through the regular expression `^de`).
This last condition is added in order to exclude from the results (22 items) those verbs that contain prefix *de-* not at the start of the word, as it is the case with *condeliquesco* 'to dissolve', where prefix *con-* appears at the start of the word before *de*.

Figure 8 shows the graph view of the derivational information connected to the node for *condeliquesco*. This figure exemplifies how the idea of linking data is realised in the LiLa Knowledge Base. The node for *condeliquesco* is connected to three nodes concerning derivational information, namely those for prefixes *de-* and *con-* and that for the Base 1266. Each of these nodes connects all the words in the lemma collection of LiLa that respectively are formed with prefix *de-* (like, for instance, *debello* 'to fight a battle (or a war) out') or suffix *con-* (e.g., *accommodo* 'to fit'), and those that share the same lexical base of *condeliquesco*, like for instance the adjective *perliquidus* 'completely fluid'.

6. Use-case Scenarios

This section presents some examples of the use of derivational data in the LiLa Knowledge Base. Examples are organised in three subsections, respectively dedicated (1) to investigations that can be performed on derivational data alone, like for instance the distribution of affixes in the lemma collection of LiLa (Subsection 6.1), (2) to complex queries on different resources interlinked in LiLa, where derivational and textual

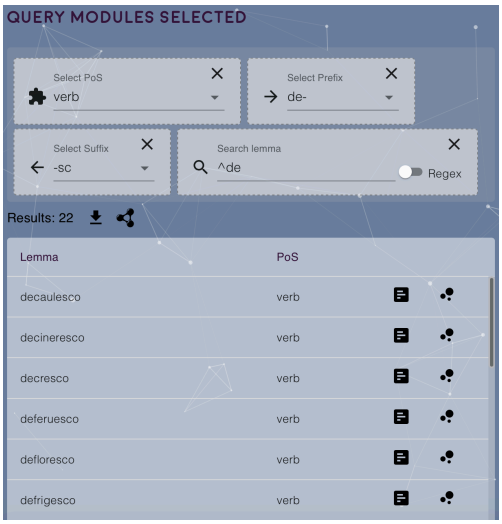


Figure 7. A complex query on derivational information in the query interface of the LiLa lemma collection.

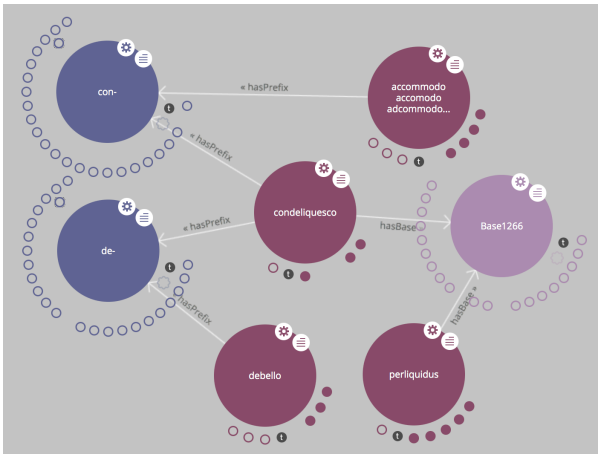


Figure 8. The graph view of lemma condeliquesco*.*

data from corpora are cross-referenced (Subsection 6.2) and (3) to the combination and comparison of two resources providing derivational information.

6.1. Inside Derivational Data

As it stands, the LiLa Knowledge Base can support a number of investigations on word formation that were not so comprehensively and instantly feasible before.

One of the most basic queries is the retrieval of all lemmas linked to the same lexical base (i.e. all the members of a word formation family) via the `hasBase` object property. The query starts by finding a given lemma, then identifies the lexical base linked to it, and finally lists all the other lemmas connected to the same base. For instance, starting from the adjective *formalis* 'of a form, formal', 67 lemmas are retrieved.[22] These can be grouped by PoS: 32 adjectives (including e.g. *serpentiformis* 'shaped like a snake' and *uniformis* 'uniform'), 25 nouns (e.g. *forma* 'shape', *formella* 'mould' and *informator* 'one who shapes'), 9 verbs (e.g. *informo* 'to shape, to inform' and *reformo* 'to transform'), and 1 adverb (*ambiformiter* 'ambiguously').

Similar queries can be performed using affixes as starting points. These can be useful, as an example, when considering that the same affixes have a tendency to be frequently associated in complex words. The LiLa Knowledge Base allows accurate empirical evidence on which among affixes are more often found together in a lemma. A query that performs this operation traverses all the lemmas in the LiLa Knowledge Base, counts all couplets of prefixes and/or suffixes, and finally reports statistics on those that are most frequently associated.

For example: with 121 instances, the most frequently associated prefixes in the LiLa lemma collection are *con-* and *in-* (with meaning of negation).[23] These two affixes are preponderantly found together in adjectives (96), such as *incommutabilis* 'unchangeable', while less frequently in nouns (23, e.g. *inconsequentia* 'lack of consistency') and adverbs (2, *incommote* 'immovably, firmly' and *incorribiliter* 'incorrigibly'). With 79 lemmas, the association of (negative) *in-* prefix and *ex-* is less frequent; examples are for instance adjective *inefficax* 'unproductive' and noun *inexperientia* 'inexperience'.

As for suffixes, the most frequent association is that of -(*i*)*t* and -(*t*)*io*(*n*), which are found in combination in 214 nouns such as *dissertatio* 'dissertation' and *excogitatio* 'a thinking out'. The second most attested combination (153 lemmas) involves again -(*i*)*t* and the suffix -(*t*)*or*, the latter mainly typical of agent or instrumental nouns. This association occurs in nouns like *dictator* 'dictator' and the adjective *gestatorius* 'that serves for carrying'.

The two most productive associations between a prefix and a suffix in LiLa are those between the negative *in-* prefix and the suffix *-bil* (296 lemmas, such as adjec-

[22] The starting word *formalis* is included in the count.

[23] In Latin there are two prefixes *in-*, one with negative and one with entering meaning.

tive *insuperabilis* 'that cannot be passed'), and between the prefix *con-* and the suffix *-(t)io(n)*, with 290 lemmas, which are mostly nouns like *contemplatio* 'viewing, contemplation' and *reconciliatio* 're-establishing'.

6.2. Outside Derivational Data

The data on word formation stored in the LiLa Knowledge Base can also be used to perform corpus-based queries. Links between lemmatised texts and the lemmas of the LiLa collection are then used to test how the different prefixes, suffixes or bases are distributed in texts.

As an example, we investigate the most frequently occurring derivational morphemes in a group of annotated textual resources, namely three Latin treebanks and one lemmatised corpus. The treebanks are the *Index Thomisticus* Treebank (IT-TB) (Passarotti, 2011), based on works written in the 13th century by Thomas Aquinas (approximately 400k nodes), the PROIEL corpus (Haug and Jøhndal, 2008), which includes the entire New Testament in Latin (the so called *Vulgata* by Jerome) along with other prose texts of the Classical and Late Antique period, and the Late Latin Charter Treebank (Korkiakangas and Passarotti, 2011) (LLCT; around 250k nodes), a syntactically annotated corpus of original VIIIth-IXth century charters from Central Italy. To those treebanks we add also the corpus of the Latin works of Dante Alighieri (13/14th century), distributed as part of the *Dante Search* project.[24]

All four resources include lemmatisation, which we use to connect the corpus tokens to the lemmas in LiLa following the procedure presented in Mambrini and Passarotti (2019). Once that the tokens in the annotated texts are linked to the LiLa lemmas, we use the SPARQL query language to extract information about the derivational morphemes attested in each corpus. While some lemmatised resources, like the IT-TB and the works of Dante, are already accessible via a dedicated endpoint provided by LiLa,[25] virtually any other lemmatised corpus can be linked and searched using local files with the methodology described in Mambrini and Passarotti (2019); the results reported here for PROIEL and LLCT were obtained by querying local files.

Table 1 reports some simple statistics on the incidence of verbs formed with the prefixes *de-* and *e(x)-* in the two corpora available in LiLa (IT-TB and Dante) and in the two treebanks queried locally (PROIEL and LLCT); in the table, we provide both the number of occurrences of any given verb formed with the two prefixes (Tokens), and of the different verbs attested (Lemmas).

The LiLa Knowledge Base can also help researchers with questions such as: what are the most frequent affixes in Latin texts? In order to observe the distribution of prefixes and suffixes in the lexicon of the PROIEL corpus, the most balanced Latin treebank in terms of textual genres, we can start from a SPARQL query that retrieves

[24] `https://dantesearch.dantenetwork.it`.

[25] `https://lila-erc.eu/sparql/corpora`.

	de-		e(x)-	
Corpus	Tokens	Lemmas	Tokens	Lemmas
IT-TB	1,013	52	1,098	76
Dante Search	299	81	379	126
PROIEL (UD)	1,011	128	1,328	152
LLCT	209	28	155	16

Table 1. Number of occurrences of verbs formed with the prefixes de- *and* e(x)- *in four corpora.*

Affix	Type	Lemmas	Tokens
-(t)io(n)	Suffix	393	2,157
con-	Prefix	344	3,297
ad-	Prefix	201	2,514
e(x)-	Prefix	197	2,713
-i	Suffix	194	2,052
de-	Prefix	182	1,294
in (entering)-	Prefix	178	1,559
-(i)t	Suffix	158	1,275
-tas/tat	Suffix	157	1,582
re-	Prefix	151	1,858

Table 2. The 10 affixes most frequently associated with a token in the PROIEL corpus.

all tokens linked with a LiLa lemma that is, in turn, connected to one or more derivational morphemes. The results are reported in Table 2. Here, while tokens of words derived with the suffix -$(t)io(n)$ rank only in the fourth place and are considerably outnumbered by tokens formed with the prefix *con-*, lemmas displaying the suffix -$(t)io(n)$ outnumber all the others; this means that, while there are more occurrences of tokens formed with *con-*, the PROIEL texts contain more words formed with -$(t)io(n)$. Such distribution reflects the greater productivity of this suffix as recorded in WFL: 2,686 lemmas formed with -$(t)io(n)$ vs. 748 with *con-*.

6.3. Comparing Derivation in Different Resources

One added value of including linguistic resources in a lexically-based Knowledge Base like LiLa, where data and metadata from distributed sources interact, is that

information about a lexical item provided by different resources can be combined and possibly compared.

Beside WFL, LiLa now includes another lexicon that provides derivational information. The Knowledge Base has been recently enriched with etymological data taken from the *Etymological Dictionary of Latin and the other Italic Languages* (de Vaan, 2008). By adopting the Ontolex-lemon model, now a *de facto* standard for the representation of lexical resources, and the lemonEty expansion designed to represent also etymological information (Khan, 2018), the etymologies were modelled to represent scientific hypotheses about the inheritance links between Latin words and the reconstructed forms of the Proto-Italic (PIt) and Proto-Indo-European (PIE) languages (Mambrini and Passarotti, 2020).

For each entry, the dictionary lists a number of derivatives of the head word in Latin, which are limited to those words whose first attestation is dated no later than the times of Cicero (106-43 BCE; de Vaan, 2008, 11-2). For instance, for the entry *donum* 'gift, present', the derivatives *donare* 'to present, give', *donabilis* 'worthy to be the recipient' and *donaticus* 'formally presented' are reported. Thus, with the inclusion in LiLa of the derivational data from WFL and of the etymological dictionary by de Vaan (2008), it becomes possible to compare (and possibly enhance) the two resources.

The comparison process starts from collecting the relevant data. We begin by selecting those lemmas of the LiLa collection that are assigned etymological information taken from de Vaan (2008). For each lemma selected, we then check whether it is connected to at least one base node in the Knowledge Base, which implies that the lemma is recorded in WFL as the member of a word formation family. We finally repeat the step for each of the Latin derivatives of the lexical entries in de Vaan (2008) included in LiLa, checking if they are present in the LiLa collection and if they are connected to at least one base node.

By using the data collected with the methodology described above, the derivatives from de Vaan (2008) are then compared to those from WFL as recorded in LiLa. This is performed in two steps:

- we calculate the number of different base nodes in LiLa connected to the derivatives listed in a lexical entry from de Vaan (2008). This informs us on whether the derivatives match the same word formation family as WFL or whether they are part of different families;
- for each word formation family in WFL, we collect all members not included among the set of derivatives of de Vaan (2008). As the derivatives in de Vaan (2008) are selected to represent only the earliest phases of the history of Latin until Cicero, this step allows us to extend the number of derivatives with words of later attestation.[26]

[26]In turn, also word formation families in WFL can be enhanced with derivatives from de Vaan (2008). There are, indeed, cases where a derivative reported by de Vaan (2008) is not recorded in WFL. In such

To give an example of how the comparison process works, in the case of the lexical entry *donum*, all the 3 derivatives reported by de Vaan (2008) are present in the LiLa Knowledge Base and all of them are connected to the same base,[27] and therefore belong to the same word formation family in WFL. Therefore, de Vaan (2008) and WFL agree that *donum* and its 3 derivatives share the same 'derivational history'. By collecting all the members of the WFL word formation family of *donum*, on the other hand, it is possible to enhance the set of derivatives reported by de Vaan (2008) with 14 further words.[28] Figure 9 shows the graphical representation of the connection of the lemmas *condonatio*, *donatiuncula*, *donarius* and *dono* which share the same base node of *donum* in LiLa.[29] Note that the lemma *donum* (with graphical variants *donom*, *dunom* and *dunum*) is connected to a Lexical Entry (*dōnum*) in de Vaan (2008) via the property `canonicalForm` and, from there, to its PIE and PIt reconstructed forms.[30].

Entries/derivatives of de Vaan (2008) with 1 base in LiLa	675
Entries/derivatives of de Vaan (2008) with 2+ bases in Lila	429
Entries of de Vaan (2008) not in WFL	14
Total de Vaan (2008) in LiLa	1,118

Table 3. Comparison between de Vaan (2008) and WFL.

Table 3 reports the number of lexical entries from de Vaan (2008) whose Latin derivatives included in WFL are connected respectively to one (675) and to two or more base nodes (429) in LiLa. Furthermore, Table 3 reports that 14 entries of the etymological dictionary are not in WFL (although 9 of them are indeed contained in the LiLa collection of lemmas).

The 675 entries of de Vaan (2008), whose derivatives included in WFL are all connected to only one base in LiLa, match those cases where the two lexical resources agree on the derivational history of the words concerned. Despite such agreement, the two resources diverge largely in the number of derivatives reported for each lex-

cases, the word in question is either absent from the LiLa collection or, if present, it is not connected to any base node (as this information should come from WFL).

[27] `https://lila-erc.eu/data/id/base/1012`.

[28] *condonatio* 'grant, donation, remission', *condonatrix* 'one who remits' (feminine), *condono* 'to deliver up, to remit', *donarium* 'temple treasury, endowment', *donarius* 'donee', *donatio* 'donation', *donatiuncula* 'small donation', *donativum* 'largess', *donator* 'donor' (masculine), *donatrix* 'donor' (feminine), *donifico* 'to make presents', *indonatus* 'unrewarded, unendowed', *redonator* 'restorer' and *redono*'to restore'.

[29] The verbal lemma *dono* corresponds to the derivative *donare* in de Vaan (2008), as in the etymological dictionary the citation form for verbs is the present infinitive instead of the first singular person of the present indicative (used in LiLa and WFL).

[30] See Mambrini and Passarotti (2020) for details.

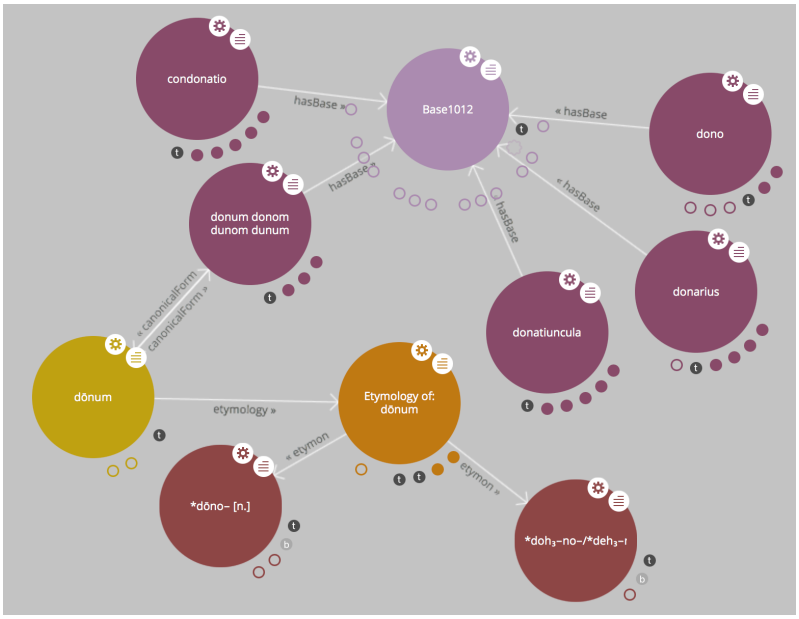


Figure 9. Lemmas connected to the same Base node of donum *in LiLa.*

ical entry or word formation family. Only 12 entries of de Vaan (2008) show a full overlap with the corresponding family in WFL, such as in the case of the adjective *sons* 'guilty', whose derivatives in both de Vaan (2008) and WFL are *insons* 'innocent' and *sonticus* 'genuine, valid'. In LiLa, these three lemmas are the only ones sharing a connection to the same base.[31]

The remaining 663 entries of de Vaan (2008) whose derivatives included in WFL are all connected to the same base node in LiLa display a different number of derivatives than those included in the same family in WFL. This happens either because a derivative reported by de Vaan (2008) is not included in WFL (and possibly in LiLa, too), or because a member of the word formation family of WFL is not reported among the derivatives for its corresponding entry in de Vaan (2008). In the former scenario, the WFL families could be enhanced with the additional data provided by de Vaan (2008); in the latter, WFL could provide candidate words to expand the range of derivational and etymological explanation provided by de Vaan (2008) with fur-

[31] https://lila-erc.eu/data/id/base/3278.

ther (and later attested) derivatives.[32] This would mean that de Vaan (2008) could contribute to the addition of 321 derivatives missing in the corresponding WFL families,[33] and that WFL could enhance the pool of the early attested derivatives in de Vaan (2008) with a total 10,438 extra lemmas.

As mentioned, it is not surprising that the number of candidate words for the inter-resource enhancement provided by WFL is much bigger than that by de Vaan (2008). While de Vaan (2008) focuses on Indo-European etymology and on the earliest stages of linguistic history, WFL aims to be as exhaustive as possible in lexical coverage, and thus includes the entire Classical Latin vocabulary well after the Republican period. Such different approach pursued by the two resources becomes an added value when these are compared and joined through a Knowledge Base like LiLa, because they provide different, yet compatible, information about the same items. Hence the added value is not only to contribute 10,438 additional lemmas to the total of derivatives reported by de Vaan (2008) in his entries, but also to obtain, for these lemmas, etymological information inherited through their connection to the same base in LiLa.

One example showing mutual enhancement is the verb *fluo* 'to flow, run (of waters)'. The lexical entry for *fluo* in de Vaan (2008) reports 26 derivatives, 23 out of which are connected to the same base in LiLa.[34] Indeed, although all the 26 derivatives of *fluo* are present in the LiLa collection, 3 of them are not recorded in WFL, which means that they are not connected to any base in LiLa. This is a case of enhancement of WFL (and, as a consequence, of LiLa, too) from de Vaan (2008), as the 3 derivatives concerned are all good candidates to be connected to the same base node of the other 23 of the same entry of de Vaan (2008). The opposite enhancement, from WFL/LiLa to de Vaan (2008), is much bigger, as there are 121 lemmas connected to the same base of *fluo* in LiLa that are not reported among its derivatives in the etymological dictionary. We manually checked that all these 121 lemmas are good candidates to be included in the list of derivatives of *fluo* in de Vaan (2008). They can all inherit the etymological information offered by the dictionary's entry.

On the other hand, there are 429 entries in the etymological dictionary whose Latin derivatives are connected to 2, or more, base nodes in LiLa (i.e. they belong to different word formation families in WFL). The reason for this falls in two main categories.

First, there can be errors in WFL, namely cases of words that must belong to the same family but are instead spread in two, or more families. Most of the cases result-

[32] We speak of 'candidate words', because each of them must be checked manually, as it cannot be taken for granted that all derivatives provided by de Vaan (2008), as well as all members of the WFL families, can be transferred from one resource to the other. However, the fact that all the derivatives of an entry of the etymological dictionary present in WFL are connected to the same base node in LiLa is a good argument in support of the portability of the information between the two resources.

[33] These 321 derivatives can be either absent from the LiLa collection, or they can be present but not connected to the base node of the WFL word formation family in question.

[34] `https://lila-erc.eu/data/id/base/183`.

ing from the 429 entries in question fall in this category. The identification of such errors must be considered a positive outcome of joining the two resources through LiLa, in that it helps to improve the quality of the connected resources. One example is the verb *eviro* 'to unman', which is listed among the derivatives of *vir* 'man' in de Vaan (2008), but it is not connected to the same base node of *vir* in LiLa,[35] and thus it does not belong to the same word formation family of *vir* in WFL. This kind of error, once discovered, can be rectified.

Second, there are cases of discrepancy due to the different perspective of the two lexical resources, reflecting the approach to word formation they pursue, their background motivation, or a different stance on the history of words. For example, in de Vaan (2008) the entry *mens* 'mind' records 8 derivatives, 7 of which match lemmas connected in LiLa to the same base of *mens*;[36] however, the noun *mentio* 'mention' in LiLa is connected to another base,[37] namely the one that connects words belonging to the WFL word formation family whose ancestor is the verb *miniscor* 'to remember'. As mentioned above, in WFL decisions on derivation are mostly based on OLD. Here the lexical entry for *mentio* is recorded as originating from the reconstructed root **men* plus suffix *-tio*, and the entries for *mens* and *miniscor* are referred to for comparison. The entry for *miniscor* states that it is cognate with *memini* 'to remember' and refers to *mentum*, i.e. the perfect participle of *miniscor*. In WFL, *mentio* is recorded as derived from *miniscor* and does not belong to the same family of *mens*, because the suffix *-tio* tends to form nouns from verbs, in particular from the base of the perfect participle of the input verb. This is exactly what happens in *mentio*, which is derived from the base of *mentum* (perfect participle of *miniscor*). In de Vaan (2008), on the other hand, *mentio* is listed as derivative in the entry of *mens*, while the verb *miniscor* is recorded as cognate with *memini*. Although the PIE words that *mens* and *memini* are derived from are etymologically related (a fact that is reflected in the cross references between the entries in the dictionary), the two are discussed under different lemmas and thus they are linked to multiple WFL families.

Table 4 sums up the recording of the words concerned in OLD, LiLa, WFL and de Vaan (2008).

7. Conclusions

In this paper, we have described the treatment of word formation in the LiLa Knowledge Base, which links together distributed linguistic resources for Latin. By reporting a number of use-case scenarios of the Knowledge Base on different issues related to derivational morphology, we have shown how helpful linguistic resources

[35] Base node of *eviro* in LiLa: `https://lila-erc.eu/data/id/base/1554`. Base node of *vir* in LiLa: `https://lila-erc.eu/data/id/base/790`.

[36] `https://lila-erc.eu/data/id/base/259`.

[37] `https://lila-erc.eu/data/id/base/961`.

Word	OLD	LiLa	WFL	de Vaan (2008)
memini	underived in Latin	Base: 2353	ancestor	head word
mens	underived in Latin	Base: 259	ancestor	head word
mentio	<**men+-tio*	Base: 961	<*miniscor*	derivative of *mens*
miniscor	cognate with *memini*	Base: 961	ancestor	derivative of *memini*

Table 4. Comparison between OLD, LiLa, WFL and de Vaan (2008) on single words.

are when they are made interoperable. Indeed, the steady work done across the last decades on new digital corpora and lexica for Latin, together with the century-long tradition of lexicography for Classical languages, has led to the current availability of a large set of linguistic resources for Latin. In different ways, all these resources concern words. For this reason, LiLa's starting point is based on the idea of linking through lemmas; each connected resource then provides its contribution to the overall picture resulting from the joining of the appropriate (meta)data from all sources.

As for derivational morphology, the information recorded in the list of Latin lemmas in LiLa is based on the WFL lexicon, which was built on the portion for Classical and Late Latin of the Lemlat lexical basis. However, since LiLa is not meant to be limited to a specific era of Latin only, extending the coverage of WFL to the Medieval Latin lemmas included in Lemlat (around 86,000) represents a major next step for the coming years. Although probabilistic models can be used in the first phase of this task (like, for instance, the one described by Sumalvico, 2017), manual disambiguation of the results, as well as the retrieval of both false positives and negatives, is to be expected.

Another potential development of the description of word formation in the LiLa Knowledge Base would be to assign some kind of linguistic information to the base nodes, which are currently just empty connectors of lemmas belonging to the same word formation family. One possible solution could be to assign to each base a written representation consisting of a string describing the lexical "element" that lies behind each lemma in the word formation family (e.g. DIC- for *dico* 'to say', or *dictio* 'a saying'). This procedure is however complicated by the fact that different bases can be used in the same word formation family: for example *fer-*, *tul-* and *lat-* can all be found as bases in the word formation family the verb *fero* 'to bring' belongs to.

Acknowledgements

This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme - Grant Agreement No 769994.

Bibliography

Bird, Steven and Mark Liberman. A formal framework for linguistic annotation. *Speech communication*, 33(1-2):23–60, 2001. doi: 10.1016/S0167-6393(00)00068-6.

Blevins, James P. *Word and paradigm morphology*. Oxford University Press, Oxford, UK, 2016. doi: 10.1093/acprof:oso/9780199593545.001.0001.

Booij, Geert. Construction morphology. *Language and linguistics compass*, 4(7):543–555, 2010. doi: 10.1093/acrefore/9780199384655.013.254.

Budassi, Marco and Eleonora Litta. In Trouble with the Rules. Theoretical Issues Raised by the Insertion of -sc- Verbs into Word Formation Latin. In *Proceedings of the Workshop on Resources and Tools for Derivational Morphology (DeriMo)*, pages 15–26. Educatt, 2017.

Budassi, Marco and Marco Passarotti. Nomen Omen. Enhancing the Latin Morphological Analyser Lemlat with an Onomasticon. In *Proceedings of the 10th SIGHUM Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities (LaTeCH)*, pages 90–94, Berlin, Germany, 2016. Association for Computational Linguistics. doi: 10.18653/v1/W16-2110.

Buitelaar, Paul, Philipp Cimiano, John McCrae, Elena Montiel-Ponsoda, and Thierry Declerck. Ontology lexicalisation: The lemon perspective. In *WS 2 Workshop Extended Abstracts, 9th International Conference on Terminology and Artificial Intelligence*, pages 33–36, 2011.

Cecchini, Flavio Massimiliano, Marco Passarotti, Paolo Ruffolo, Marinella Testori, Lia Draetta, Martina Fieromonte, Annarita Liano, Costanza Marini, and Giovanni Piantanida. Enhancing the Latin Morphological Analyser LEMLAT with a Medieval Latin Glossary. In Cabrio, Elena, Alessandro Mazzei, and Fabio Tamburini, editors, *Proceedings of the Fifth Italian Conference on Computational Linguistics (CLiC-it 2018). 10-12 December 2018, Torino*, pages 87–92. aAccademia university press, 2018. doi: 10.4000/books.aaccademia.3121.

Chiarcos, Christian. Interoperability of corpora and annotations. In *Linked Data in Linguistics*, pages 161–179. Springer, 2012. doi: 10.1007/978-3-642-28249-2_16.

Chiarcos, Christian and Christian Fäth. CoNLL-RDF: Linked Corpora Done in an NLP-Friendly Way. In Gracia, Jorge, Francis Bond, John P. McCrae, Paul Buitelaar, Christian Chiarcos, and Sebastian Hellmann, editors, *Language, Data, and Knowledge*, pages 74–88, Cham, 2017. Springer International Publishing. ISBN 978-3-319-59888-8. doi: 10.1007/978-3-319-59888-8_6. URL https://link.springer.com/content/pdf/10.1007%2F978-3-319-59888-8_6.pdf.

Chiarcos, Christian and Maria Sukhareva. OLiA - Ontologies of Linguistic Annotation. *Semantic Web Journal*, 6(4):379–386, 2015. doi: 10.3233/SW-140167. URL http://www.semantic-web-journal.net/content/olia-%E2%80%93-ontologies-linguistic-annotation.

de Vaan, Michiel. *Etymological Dictionary of Latin: and the other Italic Languages*. Brill, Amsterdam, 2008. ISBN 978-90-04-16797-1. URL https://brill.com/view/title/12612.

Declerck, Thierry, Piroska Lendvai, Karlheinz Mörth, Gerhard Budin, and Tamás Váradi. Towards linked language data for digital humanities. In *Linked Data in Linguistics*, pages 109–116. Springer, 2012. doi: 10.1007/978-3-642-28249-2_11.

Domenig, Mark and Pius ten Hacken. *Word Manager: A system for morphological dictionaries*, volume 1. Georg Olms Verlag AG, Hildesheim, 1992.

Forcellini, Egidio. *Totius latinitatis lexicon: Onomasticon ; 1 (A - B)*. Typis Aldinianis, 1867.

Georges, Karl Ernst. *Ausführliches Lateinisch-Deutsches Handwörterbuch*. Hannover: Hahn, 1913.

Glare, Peter GW. *Oxford Latin dictionary*. Clarendon Press. Oxford University Press, Oxford, UK, 1982.

Gradenwitz, Otto. *Laterculi Vocum Latinarum*. Verlag Von S. Hirzel, Leipzig, 1904.

Haug, Dag TT and Marius Jøhndal. Creating a parallel treebank of the old Indo-European Bible translations. In *Proceedings of the Second Workshop on Language Technology for Cultural Heritage Data (LaTeCH 2008)*, pages 27–34, Marrakesh, Morocco, 2008. European Language Resources Association (ELRA).

Hellmann, Sebastian, Jens Lehmann, Sören Auer, and Martin Brümmer. Integrating NLP using Linked Data. In *12th International Semantic Web Conference, Sydney, Australia, October 21-25, 2013*, 2013. doi: 10.1007/978-3-642-41338-4_7. URL `https://svn.aksw.org/papers/2013/ISWC_NIF/public.pdf`.

Hockett, Charles F. Two Models of Grammatical Description. *Words*, 10:210–231, 1954. doi: 10.1080/00437956.1954.11659524.

Ide, Nancy and James Pustejovsky. What does interoperability mean, anyway. *Toward an Operational*, 2010.

Khan, Fahad. Towards the Representation of Etymological and Diachronic Lexical Data on the Semantic Web. In McCrae, John P., Christian Chiarcos, Thierry Declerck, Jorge Gracia, and Bettina Klimek, editors, *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan, 2018. European Language Resources Association (ELRA). ISBN 979-10-95546-19-1.

Kircher-Durand, Chantal. Syntax, morphology and semantics in the structuring of the Latin lexicon, as illustrated in the -lis derivatives. In Coleman, Robert, editor, *New Studies in Latin Linguistics, Proceedings of the 4th International Colloquium on Latin Linguistics, Cambridge, April 1987*, Cambridge, 1991. John Benjamins.

Korkiakangas, Timo and Marco Passarotti. Challenges in annotating medieval Latin charters. *Journal for Language Technology and Computational Linguistics*, 26(2):103–114, 2011.

Lassila, Ora, Ralph R. Swick, World Wide, and Web Consortium. Resource Description Framework (RDF) Model and Syntax Specification, 1998.

Litta, Eleonora. On the Use of Latin -bilis Adjectives across Time. *Quaderni Borromaici. Saggi studi proposte*, 6:149–62, 2019.

Litta, Eleonora and Marco Budassi. What we talk about when we talk about paradigms. In Fernández-Domínguez, Jesús, Alexandra Bagasheva, and Cristina Lara-Clares, editors, *Paradigmatic relations in derivational morphology*. Forthcoming.

Litta, Eleonora and Marco Passarotti. (When) inflection needs derivation: a word formation lexicon for Latin. In Holmes, Nigel, Marijke Ottink, Josine Schrickx, and Maria Selig, editors, *Words and Sounds*, pages 224–239. De Gruyter, Berlin, Boston, December 2019. ISBN 978-3-11-064758-7. doi: 10.1515/9783110647587-015. URL `http://www.degruyter.com/view/books/9783110647587/9783110647587-015/9783110647587-015.xml`.

Litta, Eleonora, Marco Passarotti, and Francesco Mambrini. The Treatment of Word Formation in the LiLa Knowledge Base of Linguistic Resources for Latin. In *Proceedings of the Second International Workshop on Resources and Tools for Derivational Morphology*, pages 35–43, Prague, Czechia, September 2019. Charles University, Faculty of Mathematics and Physics, Institute of Formal and Applied Linguistics. URL https://www.aclweb.org/anthology/W19-8505.

Mambrini, Francesco and Marco Passarotti. Harmonizing Different Lemmatization Strategies for Building a Knowledge Base of Linguistic Resources for Latin. In *Proceedings of the 13th Linguistic Annotation Workshop*, pages 71–80, Florence, Italy, August 2019. Association for Computational Linguistics. doi: 10.18653/v1/W19-4009. URL https://www.aclweb.org/anthology/W19-4009.

Mambrini, Francesco and Marco Passarotti. Representing Etymology in the LiLa Knowledge Base of Linguistic Resources for Latin. In *Proceedings of the 2020 Globalex Workshop on Linked Lexicography*, pages 20–28, Marseille, France, May 2020. European Language Resources Association. URL https://www.aclweb.org/anthology/2020.globalex-1.3.

McCrae, John P., Julia Bosque-Gil, Jorge Gracia, Paul Buitelaar, and Philipp Cimiano. The OntoLex-Lemon Model: development and applications. In *Proceedings of eLex 2017*, pages 587–597, 2017. URL https://elex.link/elex2017/wp-content/uploads/2017/09/paper36.pdf.

Passarotti, Marco. Language resources. The state of the art of Latin and the *Index Thomisticus* treebank project. In Ortola, Marie-Sol, editor, *Corpus anciens et Bases de données*, number 2 in ALIENTO. Échanges sapientiels en Méditerranée, pages 301–320, Nancy, France, 2011. Presses universitaires de Nancy. ISBN 978-2-8143-0104-7.

Passarotti, Marco, Marco Budassi, Eleonora Litta, and Paolo Ruffolo. The Lemlat 3.0 Package for Morphological Analysis of Latin. In *Proceedings of the NoDaLiDa 2017 Workshop on Processing Historical Language*, pages 24–31. Linköping University Electronic Press, 2017.

Prud'Hommeaux, Eric, Andy Seaborne, et al. SPARQL query language for RDF. W3C. *Internet: https://www.w3.org/TR/rdf-sparql-query/*[*Accessed on February 27th, 2019*], 2008.

Sumalvico, Maciej. Unsupervised Learning of Morphology with Graph Sampling. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing* (*RANLP 2017*), Varna, Bulgaria, 2017. doi: 10.26615/978-954-452-049-6_093.

Tombeur, Paul. *Thesaurus formarum totius latinitatis a Plauto usque ad saeculum XXum*. Brepols, Turnhout, Belgium, 1998.

Address for correspondence:
Marco Passarotti
marco.passarotti@unicatt.it
Università Cattolica del Sacro Cuore. Largo Gemelli, 1 - 20123 Milan, Italy

PBML

The Prague Bulletin of Mathematical Linguistics
NUMBER 115 OCTOBER 2020 187–197

Focalizers and Discourse Relations

Eva Hajičová, Jiří Mírovský, Barbora Štěpánková

Charles University, Faculty of Mathematics and Physics, Institute of Formal and Applied Linguistics

Abstract

The main concern of the present contribution is the relation between the focussing function of certain particles called focalizers and the relations in discourse. We focus our attention on the English focalizers *also*, *only*, *even*, and their Czech counterparts *také*, *jenom*, *dokonce*, and base our analysis on the data from the English–Czech annotated parallel corpus PCEDT. We attempt to find out in which respects and under which conditions the selected focalizers may be said to serve in a discourse as discourse connectives and which particular discourse relations are indicated by the focalizers in question. Our analysis confirms the hypothesis that the particles *also*, *only* and *even* as well as their Czech equivalents play basically a discoursive role of explicit connectives, though in a different way and to a different extent.

1. Motivation and Research Questions

The analysis of the so-called focalizers, i.e. particles such as E. *also*, *only*, *even*, and their Czech counterparts *také*/*rovněž*/*též*/*zároveň* for *also*, *jen*/*jenom*/*pouze* for *only* and *dokonce* for *even*, based on the data from the English–Czech parallel corpus PCEDT and studied from two aspects in Hajičová and Mírovský (in prep), namely (i) their position in the sentence surface word order, and (ii) their semantic scope, has demonstrated that the intepretation of the semantic scope of these particles is highly dependent on the previous context and in several respects has an important influence on the interpretation of discourse relations. In a certain way, this issue is closely connected also to the debate on the status of these particles in the word-class system in relation to conjunctions and adverbs (see Štěpánková, 2014). These observations have led us to formulate the following two research questions:

 Corresponding author: hajicova@ufal.mff.cuni.cz
Cite as: Eva Hajičová, Jiří Mírovský, Barbora Štěpánková. Focalizers and Discourse Relations. The Prague Bulletin of Mathematical Linguistics No. 115, 2020, pp. 187–197. doi: 10.14712/00326585.011.

(i) in which respects and under which conditions the selected focalizers may be said to serve in a discourse as discourse connectives,
(ii) which particular discourse relations are indicated by the focalizers in question.

2. Data

We have based our analysis on the following data resources: (i) for Czech, the Prague Dependency Treebank of Czech (PDT 3.5, Hajič et al., 2018), containing documents of the total of about 50 thousand sentences annotated on the underlying syntactic layer also for information structure (topic–focus articulation, TFA, Hajičová et al., 1998) and containing also annotation of discourse relations (in a slightly modified PDTB style); (ii) for English, the Pennsylvania Discourse TreeBank (PDTB, ver. 2: Prasad et al., 2008, ver. 3: Prasad et al., 2019); (iii) for a comparison between Czech and English, the English–Czech parallel corpus (PCEDT, Hajič et al., 2012); (iv) the dictionary of Czech connectives (Mírovský et al., 2017; Synková et al., 2019).

We are aware that our analysis might have been influenced by the discourse genre of the annotated data of PDT and PCEDT (mostly journalistic style) but we assume that the phenomena under investigation, namely the discourse impacts of focalizers *also*, *only* and *even*, are general enough and that the genre in which they occur may have an impact only on their frequency.

3. Annotation of Underlying Syntactic and Discourse Relations

For our analysis, we have made use of the following features of the annotated data:

(a) underlying syntactic relations

The underlying layer sentence representations in the above mentioned PDT-based corpora (PDT 3.5 and in both the Czech and the English parts of PCEDT) have the form of dependency trees, with the PRED(icate) as the root of the tree corresponding to the main verb. Each node of the tree except for PRED is labeled among other features by a specification of the dependency relation (called a functor) such as ACT, PAT, ADDR, etc. (Hajič et al., 2017). One of these relations is the functor RHEM denoting the function of focalizer.

For illustration, we present in Figure 1 a dependency representation of the sentence *Only at the moment of maximum roll did I grasp what was going on.* (Czech translation: *Teprve ve chvíli největšího víření jsem pochopil, o co jde.*), where the functors printed in capitals stand for the following underlying syntactic relations: ACT for Actor, PAT for Patient (Objective), APP for Appurtenance, EXT for Manner-Extent and TWHEN for temporal modification.

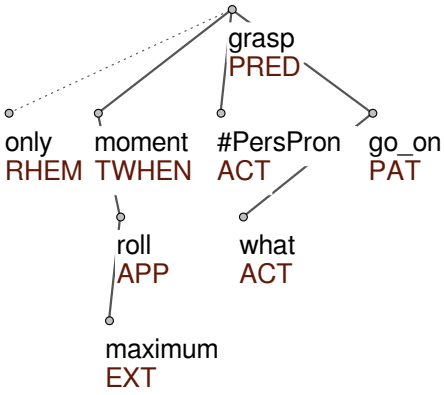


Figure 1. A simplified tectogrammatical representation of the sentence Only at the moment of maximum roll did I grasp what was going on.

(b) discourse relations

As for discourse relations, the annotation in both of these corpora is based on the Penn Discourse Treebank (PDTB) style. A discourse relation is understood to hold between two Arguments, Arg 1 and Arg 2, roughly speaking segments (adjecent sentences or in some cases between clauses within a compound sentences) including a verb as its core. The following types of relations are relevant for our discussion:[1]

(a) Explicit relation – discourse relation expressed by an explicit discourse connective (as in (1) below)

(b) Implicit relation – a certain discourse relation can be inferred but cannot be identified to be expressed by an explicit discourse connective (as in (2))

(c) EntRel – a discourse relation given by a coreference relation between entities that are a part of Arg1 and Arg2, respectively (as in (3))

(d) NoRel – no discourse relation between Arg1 and Arg2 can be recognized (as in (4))

(e) Hypophora: a coherence relation for Question-Answer pairs, where one argument (commonly Arg1) expresses a question and the other argument (commonly Arg2) provides an answer. As with Entity Relations, no explicit or implicit connective is identified and annotated.

[1] The examples are taken from the annotation manual of the PDTB 2.

Examples:

(1) *The city's Campaign Finance Board has refused to pay Mr.Dinkins $95,142 in matching funds* **<Explicit>** ***because*** *his campaign records are incomplete.*

(2) *Motorola is fighting back against junk mail. So much of the stuff poured into its Austin, Texas, offices that its mail rooms there simply stopped delivering it.* **<Implicit=so>** *Now, thousands of mailers, catalogs and sales pitches go straight into the trash.*

(3) *Pierre Vinken, 61 years old, will join the board as a nonexecutive director Nov. 29.* **<EntRel>** *Mr. Vinken is chairman of Elsevier N.V., the Dutch publishing group.*

(4) *Mr. Rapanelli met in August with U.S. Assistant Treasury Secretary David Mulford.* **<NoRel>** *Argentine negotiator Carlos Carballo was in Washington and New York this week to meet with banks.*

For the purpose of our case study we do not distinguish between a relation expressed by a one-word connective and that expressed or implied by a complex connective called AltLex, such as the one in (5):

(5) *After trading at an average discount of more than 20% in late 1987 and part of last year, country funds currently trade at an average premium of 6%.* **<AltLex>** ***The reason:*** *Shareprices of many of these funds this year have climbed much more sharply than the foreign stocks they hold.*

4. Data Analysis

4.1. Also

As for the focalizer *also*, we have searched for sentences without coordination in which *also* depends on the main verb labelled PRED. There were 1291 occurrences of this particle in the English part of PCEDT out of which 880 sentences were connected in some discourse relation with the immediately preceding sentence. As for the type of the relation, there were 828 cases annotated as an Explicit relation, 31 as an Implicit relation, 19 as an EntRel, 2 as NoRel. Out of the 828 Explicit relations there were 781 cases of the subtype Expansion.Conjunction. In 772 cases, the focalizer *also* was determined as the connective, i.e. as the indicator of the relation.

To find an answer to one of our research questions, namely whether the focalizer *also* may serve as an indicator of a certain discourse relation, we have focussed our attention on cases with *also* in which no Explicit discourse relation was annotated. There were 60 such cases in the PCEDT corpus which we have studied in relation to the preceding context. The following tendencies have been identified:

(a) In most cases, we could assign an Explicit discourse relation of the type Expansion.Conjunction, see (6).

(6) *However, excluding the year-earlier charge for recall of steering gear, operating profit in the latest quarter declined 14%, reflecting higher start-up and product development expenses in passenger-restraint systems. – Materials and production costs **also** rose, TRW said.*

(b) Only in few cases, the discourse relation EntRel could be assigned based on a coreference relation, see (7).

(7) *State Farm Mutual Automobile Insurance Co., the largest home and auto insurer in California, believes the losses from the earthquake could be somewhat less than $475 million in damages it expects to pay out for claims. – State Farm based in Bloomington, Ind, is **also** the largest writer of personal-property earthquake insurance in Calfornia.*

(c) There were also only few cases where no relation could be recognized between two adjacent sentences, see (8).

(8) *MCI has made hawks out of the upper echelon of AT&T, said T-2 PaineWebber's Mr. Grubman, who said he expected AT&T to become increasingly aggressive in dealing with longtime nemesis. – Julie Amparano Lopey in Philadelphia **also** contributed to this article.*

The statistical data quoted above and our analysis of the disputable examples has led us to the conclusion that the focalizer *also* plays a role of a connective expressing the relation of Explicit Expansion.Conjunction.

4.2. Only

The analysis of sentences containing the English focalizer *only* offers a much richer picture than those with the focalizer *also*, both as for the syntactic functions in which this particle occurs and as for the variety of Czech equivalents.

Concerning the functions the particle *only* obtains, the following four are prominent in the total of 1184 occurrences in the PCEDT (irrespective of its placement in the sentence):

- RHEM (focalizer): 750
- EXT (Extent as one of the functions of the modifier of Manner): 272
- CM (conjunction modifier): 81
- RSTR (restrictive modification of nouns, roughly speaking an attribute): 77

As for the relation of the particle *only* and the discourse relations, it should be noted that *only* serves only in 7 cases as a "pure" connective (indicating the Explicit relation of Expansion.Exception in 3 cases, of the relation Comparison.Concession in 2 cases, of the Expansion-Level-of-detail relation in 1 case and of the Comparison.Contrast in 1 case). However, there are 105 occurrences of *only* in multiword connectives (such as *not only but*, *only if*).

For the purpose of our analysis, the RHEM function is of importance, as in these cases the particle was classified by the annotators to function as a focalizer. In particular, we have been interested in cases where *only* depends on PRED and is placed before PRED so that it can be assumed that the whole predicative part of the sentence is in its scope. There were 61 such cases. After a closer inspection of these cases, only in 33 of them a discourse relation was found to hold between the sentence with *only* and the preceding sentence, the rest were sentences without such relations. Most relations were of the type Implicit (19), with only 7 Explicit ones, 5 of the type EntRel and 1 with NoRel type and 1 Hypophora. A closer look at the Implicit type has indicated that the presence of the focalizer *only* does contribute to a more detailed specification of the relation Expansion in the sense of a level of detail, see (9).

(9) *Instead, they map out a strategy in several phases from now until 1995. Most of the measures would* ***only*** *start to have an effect on beleaguered Soviet consumers in two or three years at the earliest.*

In case of an implicit relation of Comparison, the presence of the focalizer *only* contributes to the implication of a contrast, see (10).

(10) *For such products as canned vegetables and athletic shoes, devotion to a single brand was quite low, with fewer than 30% saying they usually buy the same brand.* ***Only*** *for cigarettes, mayonnaise and toothpaste did more than 60% of users say they typically stick with the same brand.*

We have also put under scrutiny those cases in which the underlying syntactic function of the particle was annotated as one of the modifications of Manner, namely EXT. In order to find out whether a presence of *only* may help to assign a particular discourse relation, we have searched for sentences in which *only*.EXT was present but which were not connected with the preceding sentence by any discourse relation. There were 76 such sentences in the PCEDT corpus. It came out that although *only* apparently does not by itself serve as a connective, its occurrence in the sentence influences the interpretation of the relation between the two adjacent sentences in a considerable way. The following tendencies have been identified:

(i) The presence of *only* indicates an explanation, more precision, substantiation, see (11) and (12).

(11) *Some even claim the group has become a lagging, not leading, indicator. The technology sector of the Dow Jones Equity Market Index has risen* ***only*** *about 6.24% this year, while the Nasdaq Composite Index has gained 18.35%.*

(12) *But the last stock market boom, in 1986, seems small compared with the current rush to market. The $6 billion that some 40 companies are looking to raise in the year ending March 31 compares with* ***only*** *$2.7 billion raised on the capital market in the previous fiscal year.*

(ii) The presence of *only* contributes to the inversion of the discourse relation, see (13).

(13) *Toyota Motor Corp.'s Lexus division also provides specifications. But **only** two-thirds of Lexus dealers are constructing new buildings according to the Lexus specs.*

(iii) The contrast is emphasized as in (14).

(14) *The number one proposal for reducing crime in the New York survey was to put more police on foot or scooter patrol, suggested by more than two-thirds of the respondents. **Only** 22% supported private security patrols funded by the merchants themselves.*

(iv) Indication of a comparison, see (15).

(15) *The U.S. Bureau of Justice Statistics reports that almost 2% of all retail-sales workers suffer injuries from crime each year, almost twice the national average and about four times the rate for teachers, truck drivers, medical workers and door-to-door salespeople. **Only** a few other occupations have higher reported rates of criminal injury, such as police, bartenders and taxi drivers.*

(v) An adversative relation is implied, see (16).

(16) *Whether psyllium makes Sidhpur's fortune depends on cholesterol-fearing Americans, the U.S. Food and Drug Administration and, of course, the outcome of further research. **Only** one thing is certain here: Psyllium is likely to remain an export item from Sidhpur for a long time.*

As mentioned above, the parallel PCEDT corpus offered a variety of Czech equivalents of the particle *only* (besides the more straightforward translations *jenom*, *jen*, *pouze*, there occurred equivalents such as *až*, *ještě*, *dokonce* or *také*) and therefore we have also looked whether the Czech translation might help to recognize a more detailed specification of the discourse relation. However, we have not found any indications in the data of such a case.

4.3. Even

The frequency of the occurrence of the particle *even* (irrespective of its position in the sentence) analyzed as a focalizer was 653 times, that is much lower than that of the focalizers *also* and a little bit lower also than that of the focalizer *only*. However, a more striking fact was that in PDTB 3 *even* does not occur as a pure connective, it occurs only as a part of some multiword complex connectives such as *even if*, *even though*, *even as*, *even when* etc.

Therefore we have looked in more detail at the Czech translations of this particle to see if the Czech translations in the given contexts may offer a more varied picture. We have found 19 different Czech equivalents of *even*.RHEM, the most frequent of which was *dokonce* (242 times) and *ještě* (113 times).

Having these data at our disposal, we have decided to investigate whether the occurrence of *even*.RHEM translated as *dokonce* may influence the discourse relations, that is to say if it may play a role of a true connective. We have focussed our attention on the position of *even*.RHEM before the PRED (in non-coordinated constructions) and translated as *dokonce*, which occurred 98 times. Out of this number, there were 65 cases where a discourse relation to the previous sentence was annotated, 54 of which were marked as Implicit relations (of the type Expansion.Conjunction 32, other type of Expansion 14 and other Implicit 8); there were 8 Explicit relations (of the type Expansion.Conjuction 2, Comparison.Concession 4, Comparison.Contrast 1, and Temporal.Asynchronous 1), 2 relations were marked as EntRel and 1 as AltLex. None of the Explicit relations was marked by the focalizer *even*, the connectives were *but* (3), *and* (2), *however*, *still*, *even then*.

Looking at the Implicit relations in more detail, we have seen that in most cases marked as Expansion, there was a certain degree of gradation involved, see e.g. (17) and (18) with Expansion.Conjunction marked as "in fact". The same is true with the relation annotated as Comparison.Concession and marked in as "nevertheless" in (19).

(17) *All kinds of landmark Texas real estate has been snapped up by out-of-staters.* ***Even*** *the beloved Dallas Cowboys were bought by an Arkansas oil man.*

(18) *Mr Hahn began selling non-core businesses, such as oil and gas and chemicals. He* ***even*** *sold one unit that made vinyl checkbook covers.*

(19) *But that's for the best horses, with most selling for much less.* ***Even*** *when they move outside their traditional tony circle, racehorse owners still try to capitalize on the elan of the sport.*

Also in case of an Explicit relation one can recognize a certain gradation, see e.g. (20) annotated as Expansion.Conjunction with the connective *and*:

(20) *Press agents and public-relations practitioners are notorious name-droppers. And some* ***even*** *do it with malice afterthought.*

Our analysis of the interpretation of discourse relations between sentences the second of which contains the focalizer *even* has led to a proposal to introduce into the set of connectives the particle *even* for those relations of Expansion (and perhaps also of Comparison) that can be interpreted as gradation. It should be noted that the type gradation is not among the types of relations recognized by PDTB. Such a solution would comply with the treatment applied in the PDT, namely taking "dokonce" as a connective present in the relation of gradation (73 cases in total).

5. Conclusion and Summary

In the present case study, we have carried out an analysis of discourse relations between adjacent sentences (taken as discourse arguments) the second of which (ARG2)

contained one of the particles *also*, *only* or *even* in the (underlying syntactic) function of a focalizer (RHEM). Our analysis was based on the data from the annotated Czech–English parallel PCEDT and in the classification of the discourse relations we used basically the PDTB approach.

The statistical data quoted above and our analysis of the disputable examples has led us to the conclusion that the particle *also* as well as its Czech equivalents functions as a focalizer and plays basically a discoursive role of an explicit "pure" connective expressing the relation of Expansion.Conjunction.

As for the particle *only*, the PCEDT data indicate that the prevalent underlying syntactic function of this particle is that of a focalizer and of a modification of Manner. In contrast to the focalizer *also*, the particle *only* serves as a "pure" discourse connective only in a negligible number of cases, relatively more frequently being a part of a multiword connective. However, the presence of this particle helps to understand a given discourse relation in a more specific way, for instance in the sense of a certain level of detail with the relation Expansion. With the relation of Comparison, the presence of the focalizer *only* implies a contrast. If *only* obtains the function of a modification of manner EXT, it contributes to the interpretation of the relation between two neighbouring sentences in a considerable way as well, strengthening a contrastive interpretation of this relation, indicating a comparison or a relation based on coreferential entities occurring in the two sentences.

A most interesting case is offered by the analysis of the focalizer *even*. Since it does not appear in the PDTB list of connectives (it occurs only as a part of some multiword complex connectives such as *even if*, *even though*, *even as*, *even when*), we have looked for the most frequent Czech equivalent of *even*.RHEM, namely *dokonce*, and considered its possible influence on the discourse relations, that is to say we wanted to find out if it may play a role of a true connective. In most cases the relation to the previous sentence was annotated as an Implicit Relation (mostly Expansion.Conjunction). A closer inspection of these examples has led to a recognition of a certain degree of gradation present and to a conclusion that the focalizer even may be understood as a connective with this meaning.

In a follow-up analysis of the data of PCEDT we want to include into our consideration other English focalizers as candidates for the role of connectives (e.g. *mainly*, *just*), investigate them in relation to their Czech translations and thus to analyze the role of focalizers as connectives in a broader perspective than allowed by our present case study.

Acknowledgements

We gratefully acknowledge support from the Grant Agency of the Czech Republic, project 20-09853S. The work described herein has been using resources provided by the LINDAT/CLARIAH-CZ Research Infrastructure, supported by the Ministry of Education, Youth and Sports of the Czech Republic (project no. LM2018101).

This paper is a full version of the contribution presented at the workshop on discourse markers at the 2020 annual conference of Societas Linguistica Europaea. Only a short abstract of the contribution was published in the electronic version of the conference proceedings.

Bibliography

Hajič, Jan, Eduard Bejček, Alevtina Bémová, Eva Buráňová, Eva Hajičová, Jiří Havelka, Petr Homola, Jiří Kárník, Václava Kettnerová, Natalia Klyueva, Veronika Kolářová, Lucie Kučová, Markéta Lopatková, Marie Mikulová, Jiří Mírovský, Anna Nedoluzhko, Petr Pajas, Jarmila Panevová, Lucie Poláková, Magdaléna Rysová, Petr Sgall, Johanka Spoustová, Pavel Straňák, Pavlína Synková, Magda Ševčíková, Jan Štěpánek, Zdeňka Urešová, Barbora Vidová Hladká, Daniel Zeman, Šárka Zikánová, and Zdeněk Žabokrtský. *Prague Dependency Treebank 3.5*. Data/Software, Univerzita Karlova, MFF, ÚFAL, Prague, Czech Republic, 2018.

Hajič, Jan, Eva Hajicová, Jarmila Panevová, Petr Sgall, Ondrej Bojar, Silvie Cinková, Eva Fucíková, Marie Mikulová, Petr Pajas, Jan Popelka, Jiří Semecký, Jana Šindlerová, Jan Štěpánek, Josef Toman, Zdeňka Urešová, and Zdeněk Žabokrtský. Announcing Prague Czech–English Dependency Treebank 2.0. In *LREC*, pages 3153–3160, 2012.

Hajič, Jan, Eva Hajičová, Marie Mikulová, and Jiří Mírovský. *Prague Dependency Treebank*. Springer Verlag, Berlin, Germany, 2017. ISBN 978-94-024-0879-9.

Hajičová, Eva, Barbara Hall Partee, and Petr Sgall. *Topic–Focus Articulation, Tripartite Structures, and Semantic Content*. Dordrecht: Kluwer, 1998.

Hajičová, Eva and Jiří Mírovský. Focalizers through the Lens of a Parallel English–Czech Corpus. In *Submitted to COLING 2020*, in prep.

Mírovský, Jiří, Pavlína Synková, Magdaléna Rysová, and Lucie Poláková. CzeDLex – A Lexicon of Czech Discourse Connectives. *The Prague Bulletin of Mathematical Linguistics*, 109(1):61–91, 2017.

Prasad, Rashmi, Alan Lee, Nikhil Dinesh, Eleni Miltsakaki, Geraud Campion, Aravind Joshi, and Bonnie Webber. *Penn Discourse Treebank Version 2.0*. Data/Software, Linguistic Data Consortium, 2008. University of Pennsylvania, Philadelphia. LDC2008T05.

Prasad, Rashmi, Bonnie Webber, Alan Lee, and Aravind Joshi. *Penn Discourse Treebank Version 3.0*. Data/Software, Linguistic Data Consortium, 2019. University of Pennsylvania, Philadelphia. LDC2019T05.

Štěpánková, Barbora. *Aktualizátory ve výstavbě textu, zejména z pohledu aktuálního členění* [*Focalizers in the structure of text, esp. from the point of view of information structure*]. Ústav formální a aplikované lingvistiky, 2014.

Synková, Pavlína, Lucie Poláková, and Jiří Mírovský. *CzeDLex 0.6*. Data/Software, LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University, 2019. http://hdl.handle.net/11234/1-3074.

Address for correspondence:
Eva Hajičová
hajicova@ufal.mff.cuni.cz
Institute of Formal and Applied Linguistics
Faculty of Mathematics and Physics, Charles University
Malostranské náměstí 25
118 00 Praha 1
Czech Republic

The Prague Bulletin of Mathematical Linguistics
NUMBER 115 OCTOBER 2020

INSTRUCTIONS FOR AUTHORS

Manuscripts are welcome provided that they have not yet been published elsewhere and that they bring some interesting and new insights contributing to the broad field of computational linguistics in any of its aspects, or of linguistic theory. The submitted articles may be:

- long articles with completed, wide-impact research results both theoretical and practical, and/or new formalisms for linguistic analysis and their implementation and application on linguistic data sets, or
- short or long articles that are abstracts or extracts of Master's and PhD thesis, with the most interesting and/or promising results described. Also
- short or long articles looking forward that base their views on proper and deep analysis of the current situation in various subjects within the field are invited, as well as
- short articles about current advanced research of both theoretical and applied nature, with very specific (and perhaps narrow, but well-defined) target goal in all areas of language and speech processing, to give the opportunity to junior researchers to publish as soon as possible;
- short articles that contain contraversing, polemic or otherwise unusual views, supported by some experimental evidence but not necessarily evaluated in the usual sense are also welcome.

The recommended length of long article is 12–30 pages and of short paper is 6–15 pages.

The copyright of papers accepted for publication remains with the author. The editors reserve the right to make editorial revisions but these revisions and changes have to be approved by the author(s). Book reviews and short book notices are also appreciated.

The manuscripts are reviewed by 2 independent reviewers, at least one of them being a member of the international Editorial Board.

Authors receive a printed copy of the relevant issue of the PBML together with the original pdf files.

The guidelines for the technical shape of the contributions are found on the web site `http://ufal.mff.cuni.cz/pbml`. If there are any technical problems, please contact the editorial staff at `pbml@ufal.mff.cuni.cz`.