The Prague Bulletin of Mathematical Linguistics
NUMBER 110 APRIL 2018

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 110 APRIL 2018

CONTENTS

PBML

The Prague Bulletin of Mathematical Linguistics
NUMBER 110 APRIL 2018

EDITORIAL

The Editorial Board of the Prague Bulletin of Mathematical Linguistics deeply regrets to announce that we have lost a most respectful member of the Board, Professor Aravind Joshi.

Aravind K. Joshi (born August 5, 1929, died December 31, 2017), the Henry Salvatori Professor Emeritus of Computer and Cognitive Science, a founding co-director of the former Institute for Research in Cognitive Science (IRCS) at the University of Pennsylvania and a recipient of numerous honors and awards (such as Honorary Doctorate of Charles University in Prague – 2013, Benjamin Franklin Medal in Computer and Cognitive Science of the Franklin Institute – 2005, Cognitive Science Society David Rumelhart Prize – 2003, ACL Lifetime Achievement Award – 2002, NAE Member – 1999, ACM Fellow – 1998, Founding Fellow AAAI – 1990, IEEE Fellow – 1976), has been a distinguished member of the whole research community, highly appreciated for his intellectual curiosity and his enthusiasm. He has been an inspiration for dozens of his PhD students and colleagues working all over the world.

The scope of his own research interests was very wide, covering the field of Computational Linguistics, Cognitive Science and Artificial Intelligence, paying due attention to the issues at the intersection of these fields and with extensions beyond these fields (cf. e.g. his long-time interest in macromolecular structures).

Our first face-to-face meeting with Professor Joshi was at COLING in 1969 in Sweden and we have been in contact since then, as much as the political restrictions in our country allowed. After the positive political changes in Central and Eastern Eu-

rope in 1989, the contacts intensified and we have been meting regularly in Prague, Philadelphia and at conferences all over the world. Since the time he joined the Editorial Board of the Prague Bulletin, our professional contacts have become even more intensive and we have always appreciated his advice, suggestions and initiative.

There are at least three particular points of intersection of professional interests between his scholarly work and research carried out at Charles University, Prague, Czech Republic, some of which go as back as to the late sixties.

In those times Petr Sgall and his collaborators in Prague formulated an original type of generative description of language, the Functional Generative Description. One of the important issues was the discussion of the generative power of such a description. For us, Aravind's work on the so-called mildly context-sensitive grammar formalism was most inspirational and supportive because the formalism developed in our group was very close to such a concept.

Later on, when Professor Joshi formulated his Tree Adjoining Grammar formalism, we have profited much from his insights into the relation between his formalism and the dependency grammar we subscribe to.

Last but not least, and most important especially for the young team working on discourse, was his elucidation and application of Centering Theory and the build-up and development of the Penn Discourse Treebank. The content of this work laid the foundations for the Czech-American collaborative project on discourse analysis and annotation and offered an unforgettable opportunity for us to put our hands, so to say, on the Penn Discourse Treebank during our trips to Philadelphia, and to enjoy his and his colleagues' visits to Prague.

Professor Aravind Joshi was a great scientist, a wonderful teacher and a remarkable personality. His passing away is a great loss for the whole community. Computational linguists will miss him terribly as a respectful scientist, innovator in many areas, a mentor of young colleagues and students, and most of us also as a very modest, kind, calm and charming friend.

While we all will miss him, his everlasting kind smile will stay in our memories forever.

The PBML

PBML

The Prague Bulletin of Mathematical Linguistics
NUMBER 110 APRIL 2018 7–42

Modelling Morphographemic Alternations in Derivation of Czech

Magda Ševčíková

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

Abstract

The present paper deals with morphographemic alternations in Czech derivation with regard to the build-up of a large-coverage lexical resource specialized in derivational morphology of contemporary Czech (DeriNet database). After a summary of available descriptions in the Czech linguistic literature and Natural Language Processing, an extensive list of alternations is provided in the first part of the paper with a focus on their manifestation in writing. Due to the significant frequency and limited predictability of alternations in Czech derivation, several bottom-up methods were used in order to adequately model the alternations in DeriNet. Suffix-substitution rules proved to be efficient for alternations in the final position of the stem, whereas a specialized approach of extracting alternations from inflectional paradigms was used for modelling alternations within the roots. Alternations connected with derivation of verbs were handled as a separate task. DeriNet data are expected to be helpful in developing a tool for morphemic segmentation and, once the segmentation is available, to become a reliable resource for data-based description of word formation including alternations in Czech.

1. Introduction

Concerning the internal structure of complex words in the Czech lexicon, derivation is the dominant process of word formation, highly prevailing over compounding in Czech (Dokulil, 1962; Dokulil et al., 1986). All types of derivation (esp. prefixation and suffixation) in Czech may be accompanied by vowel or consonant alternations in the root and/or in affixes.[1] Morphographemic alternations are the major source

[1] In the paper, the term "root" refers to a morpheme that cannot be further analysed while "stem" is used, less specifically, for the part of a word without inflectional affixes (Haspelmath and Sims, 2010; Aronoff,

 Corresponding author: sevcikova@ufal.mff.cuni.cz
Cite as: Magda Ševčíková. Modelling Morphographemic Alternations in Derivation of Czech. The Prague Bulletin of Mathematical Linguistics No. 110, 2018, pp. 7–42. doi: 10.2478/pralin-2018-0001.

of allomorphy in Czech. They diversify the formal shape of a base word and the particular derived word; cf. palatalization of the final consonant of the root morpheme by adding a diminutive suffix in ex. (1) and an analogous alternation of the final consonant in the first diminutive suffix during the subsequent formation of a double diminutive in (2). Several alternations in a single derivational step are documented in ex. (3), namely a vowel alternation in the prefix and a consonant alternation in the final position of the root, or in (4) with a vowel alternation, a vowel insertion, and a consonant alternation in the root.

(1) *hroch*$_{\mathrm{N}}$ 'hippo' $\xrightarrow{\text{ch>š}}$ *hroš-ík*$_{\mathrm{N}}$ (dimin.)[2]

(2) *hroš-ík*$_{\mathrm{N}}$ 'hippo' (dimin.) $\xrightarrow{\text{k>č}}$ *hroš-íč-ek*$_{\mathrm{N}}$ (double dimin.)

(3) *vy-skoč-i-t*$_{\mathrm{V}}$ 'to leap' $\xrightarrow{\text{y>ý, č>k}}$ *vý-skok*$_{\mathrm{N}}$ 'leap'

(4) *vejc-e*$_{\mathrm{N}}$ 'egg' $\xrightarrow{\text{e>a, 0>e, c>č}}$ *vaječ-ný*$_{\mathrm{A}}$ 'made from eggs'

The paper is organized as follows. Starting with a note on terminology, Section 2 provides an overview of linguistic descriptions of morphographemic alternations and available approaches in Natural Language Processing (NLP) of Czech, including the derivational database DeriNet which is in focus of the paper. A detailed classification of alternations in the contemporary Czech lexicon follows in Section 3. Attached to the section, we provide a complete list of alternations supported with examples.

Due to the size of the DeriNet database (exceeding 1 million words), derivational relations, including all types of alternations, have been identified semi-automatically (Sect. 4). Suffix-substitution rules proved to be efficient for alternations in the final position of the stem, whereas a specialized approach of extracting alternations from inflectional paradigms was used for modelling alternations within the roots. Alternations connected with derivation of verbs were handled as a separate task.

Section 5 concludes with an analysis of main types of alternations not yet covered in DeriNet and provides a perspective of using the DeriNet data in the development of a tool for morphemic segmentation as well as in the linguistic research into word formation in general and into morphographemic alternations in particular.

1994). Roots and stems are not together referred to as "bases" (cf. Bauer, 1983, pp. 20f) since we reserve the term "base" for the opposition of a base word vs. a derived word (target word, or derivative). These pairs are referred to as "pairs of base-target words" or "base-target pairs", too.

[2]In the examples, the base word is written first followed by the derivative, the derivational relation is represented by an arrow. The alternations that accompany the derivation are listed above the arrow. The grapheme in the base is written first followed by ">" and the corresponding grapheme in the derivative. Boundaries between morphemes are indicated with the hyphens (the morphemic structure is not marked in Sect. 4 since the data are not segmented in the DeriNet network).
In examples on diminutive derivation, we use "dimin." (diminutive) and "double dimin." (double diminutive) instead of the full English translation (e.g. 'small hippo' and 'very small hippo', respectively).

2. Related work

2.1. Note on terminology

Unlike the (mainstream) phonemic and phonological approach of alternations (e.g. Haspelmath and Sims, 2010), the present paper deals with this issue in relation to written Standard Czech.[3] The term "morphographemic alternations" is thus preferred to that of "morphophonemic alternations" or similar terms used in the linguistic literature (cf. morphonological alternations / morphophonological alternations in Matthews, 2007, p. 253; Štekauer et al., 2012; Osolsobě, 2014, pp. 198ff; Ziková, 2015, 2016a,b; Šefčík, 2016b, or phoneme alternations / phonemic alternations / phonological alternations / alternations of phonemes in Dokulil, 1962; Daneš et al., 1967; Dokulil et al., 1986; Osolsobě, 2002; Aronoff, 1976). We neither use the term "ablaut" nor "apophony" (e.g. Lieber and Štekauer, 2014, pp. 125f, Baerman, 2015), since the former term is delimited inconsistently in the description of Czech and the latter term is not anchored in the Czech terminology; see Šefčík (2016a) for details.

Our approach is rooted in Dokulil's onomasiological theory (Dokulil, 1962) and uses common terminology on general aspects of word formation. Lexemes that share the root are called a derivational family; if members of a derivational family are organized according to the direct derivational relations, we speak about derivational trees with regard to the derivational data, rather than using Dokulil's term "word-formation nest".[4] A comment is required on the term "word-formation type" which is defined as a set of words that share a certain word-formation meaning and were derived from bases of the same part-of-speech category by using the same affix (Dokulil, 1962, pp. 68ff); cf. the word-formation type of agentive nouns derived from verbs with the suffix *-tel* in Czech (*učitel* 'teacher', *pozorovatel* 'observer').[5]

2.2. Descriptions of alternations in linguistic literature on Czech derivation

Morphographemic alternations in Czech originate in systemic as well as accidental diachronic changes that emerged during differentiation of Czech from other Slavic languages and are thus subject to historical grammars of the Czech language (Gebauer, 1984–1929; Lamprecht et al., 1986). Consonant alternations were described as changes of non-palatal consonants into palatalized ones (i.e. palatalization; or vice versa as

[3] Alternations that are not mirrored in writing are omitted in the paper; esp. palatalization of consonants is often recorded by the letters *i* or *ě* following the consonant instead of changing the grapheme itself (cf. *t–t(i)* in *bota* 'shoe' → *botička* (dimin.) instead of *t>ť* according to the pronounciation).

[4] The term "word-formation nest" was substituted for the term "derivational paradigm" by Dokulil et al. (1986, p. 207); the latter term has recently established as the core concept of the paradigmatical approach to word formation (Lieber and Štekauer, 2014, pp. 354ff; Booij, 2008; Pounder, 2000; Bauer, 1997 etc.).

[5] Dokulil's definition of the word-formation type ("slovotvorný typ") is thus different from that by Hansen (1985, pp. 28ff) ("Wortbildungstyp").

depalatalization), mostly due to the contact with a front or iotified vowel in order to allow for a more comfortable pronunciation. Three rounds of palatalization of velar consonants in Proto-Slavic were reconstructed, for each round several irregularities and exceptions were stated (Lamprecht et al., 1986; Večerka, 2016). The source of a part of vowel-zero alternations in contemporary Czech are both systematic and accidental changes of the yer-vowels in Proto-Slavic (Lamprecht et al., 1986; Ziková, 2016b). However, due to different counter-tendencies, such as the trend to preserve the vowel quantity of the base word; cf. the *e*>*é* alternation in ex. (5) vs. its lack in (6), and *o*>*ů* in (7) vs. (8) (Dokulil, 1962, p. 170), the resulting synchronic picture of alternations in the Czech derivation seems to be highly irregular (similarly to other languages; cf. Bybee and Brewer, 1980).

(5) *oheň*$_{\text{N}}$ 'fire' $\xrightarrow{\text{e>é, ň>n}}$ *ohén-ek*$_{\text{N}}$ (dimin.)

(6) *účes*$_{\text{N}}$ 'hairstyle' → *účes-ek*$_{\text{N}}$ (dimin.)

(7) *krok*$_{\text{N}}$ 'step' $\xrightarrow{\text{o>ů, k>č}}$ *krůč-ek*$_{\text{N}}$ (dimin.)

(8) *blok*$_{\text{N}}$ 'block' $\xrightarrow{\text{k>č}}$ *bloč-ek*$_{\text{N}}$ (dimin.)

Diachronic changes and the synchronic distribution of vowel-zero alternations in Czech were treated within the framework of generative phonology (Scheer et al., 2011; Scheer and Ziková, 2010). The diachronic perspective is also taken by Pognan and Panevová (2013) who examine common Slavic roots as a basis for research into Slavic intercomprehension. Less recent studies (Stankiewicz, 1986, 1960; Rubenstein, 1950) placed alternations in Czech in the cross-linguistic context of other Slavic languages.

A synchronic description of alternations in derivation was included in Dokulil's fundamental study on Czech derivation (Dokulil, 1962, esp. pp. 159–178), which has become a widely respected and, in fact, the only common ground of word-formation descriptions in Czech grammars and specialized studies since then (Daneš et al., 1967; Dokulil et al., 1986; Čermák, 2012; Štícha, 2013 etc.).[6] However, description of alternations is usually spread over the chapters on word formation and inflectional morphology with only sporadic mutual links. The most complex and elaborate description so far is by Ziková (2015), which is still a pilot study for an intended grammar of Czech and is limited to quantitative alternations and vowel-zero alternations.

Two existing morphemic dictionaries might be relevant for the topic of alternations in Czech. In Šiška's dictionary (Šiška, 2005), root morphemes of a part of the Czech lexicon are grouped together according to their lexical meaning; each n-tuple of the root allomorphs is supplemented with a selective list of lexemes. In the dictionary by

[6]Dokulil's approach, based on differentiation of four onomasiological categories, has influenced approaches to derivation in Czech as well as in other, particularly (but not exclusively) Slavic languages; cf. works on Slovak (e.g. Buzássyová, 1974; Horecký et al., 1989; Furdík, 2004), Polish (Grzegorczykowa et al., 1998), Russian (Švedova, 1980), or Štekauer's application to English (Štekauer, 1998).

Slavíčková (1975), lexemes are analysed into morphemes and listed retrogradely without mutual connections. None of the dictionaries is machine tractable, their usability for our task was very limited.

A formalized description of selected types of alternations in Czech inflection was a part of the inflectional dictionary by Osolsobě (1996); it focused on alternations of consonants in the final position of the stem. The dictionary was used in automatic morphological analysis by the Ajka (later on, Majka) analyser and in other tasks in NLP of Czech (Osolsobě, 2015); see Sect. 2.3.

2.3. Alternations in Natural Language Processing and language resources for Czech

In NLP of Czech, alternations were handled in both large-coverage inflectional analysers used for Czech, namely in the Ajka analyser (Sedláček and Smrž, 2001; Sedláček, 2004; Šmerk, 2007) and in the analyser developed by Hajič (2004).

The dictionary of the analyser Ajka can be searched for derivationally related pairs (or n-tuples) by the tool Deriv (Osolsobě et al., 2009) using regular expressions. When searching for pairs of words with alternations, each alternation must be specified with a separate regular expression. A similar tool, Morfio (Cvrček and Vondřička, 2013), searches for pairs with a common base and different affixes in the Czech National Corpus; the words need not to be in a derivational relation. The tool makes it possible to include several tens of pairs of alternations into the queries (44 pairs without respect to which of the graphemes is in the base and in the derivative). However, both tools suffer from massive overgeneration.

In a close relation to Ajka, a derivational analyser for Czech called Derivancze was developed (Pala and Šmerk, 2015). The data of Derivancze are not available for a free download, but can be queried by a web tool. For a word filled in into the web form, the tool gives a base word and a direct derivative if found in the underlying dictionary data. It was not explicitly addressed by Pala and Šmerk (2015) whether and to which extent alternations were handled in Derivancze. Nevertheless, a random search for several examples containing alternations showed a rather unsystematic approach to this phenomenon. For instance, the diminutive *domek* is correctly linked with the base noun *dům* 'house' in Derivancze whereas the diminutive *stolek* is connected incorrectly with a non-existing string *stol* (instead of *stůl* 'table'), *hrošík* 'hippo' (dimin.) was not found by the tool, no parent was found for *chirurzka* 'woman surgeon'.

The morphological analyser by Hajič is connected with the inflectional dictionary MorfFlex CZ (Hajič and Hlaváčová, 2013). From MorfFlex CZ, the set of lexemes for the DeriNet database was extracted and, moreover, the dictionary has turned out to be an important source of information on morphographemic alternations in derivation; see Sect. 4.[7]

[7] MorfFlex CZ (and thus DeriNet) covers the entire lexicon of contemporary Czech including proper nouns, archaic words, low-frequency words and regular, automatically generated coinages without respect to whether they are attested in a corpus.

Derivational relations are included in other language resources, too, though rather marginally. In Czech WordNet a set of 14 relations was implemented (Pala and Smrž, 2004; Pala and Hlaváčková, 2007). In the Prague Dependency Treebank (Hajič et al., 2006), selected types of derivatives were represented by the lemma of their base word within the deep-syntactic annotation (Razímová and Žabokrtský, 2006).

2.4. DeriNet database as a resource specialized in Czech derivation

A decision that we had to make at the start of the DeriNet project was whether pairs of base and target words with alternating graphemes will be linked together in the database, or whether they stay unconnected. The insufficient attention paid to alternations in Czech linguistics and in NLP of Czech in combination with the complicated nature of alternations were strong arguments against the inclusion of this issue into the semi-automatically constructed resource. On the other side, massive presence of alternations was the main argument in favour of including them into the database.

DeriNet is a large-coverage lexical resource specialized in derivational morphology of Czech; neither composition nor combined word-formation processes have been included so far. It is the only one freely available derivational resource for Czech and, in a broader context of European linguistics, it is in line with recent research in word formation; e.g. word-formation database for Latin (Litta et al., 2016), Démonette for French (Hathout and Namer, 2014), the language-independent approach by (Baranes and Sagot, 2014), DerivBase.Hr for Croatian (Šnajder, 2014), DerivBase for German (Zeller et al., 2013), or CELEX for English, German and Dutch (Baayen et al., 1995).[8]

The design of DeriNet was based on Dokulil's understanding of word-formation nests as internally structured groups of all words based both formally and semantically[9] on the same base in contemporary language without regard to their real etymology (Dokulil, 1962, p. 14, Dokulil et al., 1986, p. 207). Words (represented as nodes in DeriNet) are connected with a link (edge) if they are derivationally related; the edge is oriented from the base to the derivative. At most one base word may be identified for a derived word. Words that are directly and indirectly derived from a particular base word thus form an oriented graph (called derivational tree in the paper).

[8] Approach to alternations was mostly not addressed in the respective publications. Alternations are explicitly referred to by Šnajder (2014), whereas they were not included e.g. by Baranes and Sagot (2014).

[9] The formal and semantic relations of a derived word to its base are discussed as foundation and motivation, respectively, in the onomasiological theory of word-formation (Dokulil, 1962; Dokulil, 1994, pp. 131ff; Štekauer, 1998). If foundation is not in accordance with motivation, priority is given to formal relations (foundation).

The current version of the database, DeriNet 1.4,[10], contains nearly 1,012 thousand lexemes which were extracted from the MorfFlex CZ dictionary. The lexemes are interconnected with more than 774 thousand derivational links.[11] All types of alternations described in Sect. 3 have been included into DeriNet; the methods used are described in Sect. 4.

3. Morphographemic alternations in derivation of Czech

3.1. Delimitation of alternations, basic classification

An alternation is understood as a substitution of a grapheme by another one that occurred during derivation in addition to the proper affixation; the term is used both for the process of replacing a grapheme with another one in a particular morphosyntactic context and for the pair of graphemes occurring in a particular position of the base word and the target word, i.e. for the result of this process. The alternations are identified in a morpheme that is shared by the base and the derivative; cf. ex. (1) to (8) above. On the contrary, examples in (9) and (10) are not considered to contain alternations, the difference *í* vs. *i* in (9) being interpreted as a result of replacing the masculine suffix by the feminine one (resuffixation; Šimandl, 2016), and *a* vs. *á* in (10) as resulting from the substitution of the inflectional ending for a suffix.

(9) *tanečn-ík*$_{\mathrm{N}}$ ‘dancer’ → *tanečn-ice*$_{\mathrm{N}}$ ‘female dancer’

(10) *brank-a*$_{\mathrm{N}}$ ‘goal’ → *brank-á̌r*$_{\mathrm{N}}$ ‘goalkeeper’

A grapheme alternates with another grapheme or with one of a closed set of graphemes; e.g. *c* changes into *k* in (11) or into *č* in (12). Due to this feature, Osolsobě (2002) describes alternations as a “regular” substitution. It is stressed, however, that the alternations are regular neither in the sense that a given grapheme is always affected by alternation in the given morphographemic context (see (5) vs. (6), and (7) vs. (8)), nor that they are related to a particular type of derivation (e.g. defined by the part-of-speech categories of the base and target word) or even to a particular word-formation type. For instance, the *c*>*č* alternation occurs in derivation of deverbal nouns (12) and in derivation of adjectives from nouns (13). In (14) the *a*>*á* alternation must be applied, otherwise the adjective *vratný* ‘returnable’ might be connected incorrectly with the noun *vrata* ‘gate’ (but it belongs to *vrátit* ‘to return’ with the reverse alternation *á*>*a* in (15)). In (16), the alternation is not present – if applied, the adjective *slávistický* ‘belonging to supporters of Slávie’ in (17) would be derived incorrectly.

[10] `http://ufal.mff.cuni.cz/derinet`
DeriNet 1.0 and 1.2 were published in the Lindat/Clarin repository (Vidra et al., 2015, 2016). The data are freely available for non-commercial purposes under the Creative Commons (CC-BY-NC-SA) licence.

[11] For 238 thousand (23.5 % out of all nodes) no base word has been identified so far. However, more than a half of the parentless nodes is capitalized nouns (more than 124 thousand). Capitalization concerns proper nouns only, which have a limited derivational potential.

(11) *péc-t*$_{\mathrm{V}}$ ‘to bake’ $\xrightarrow{\text{é>e, c>k}}$ *pek-ař*$_{\mathrm{N}}$ ‘baker’

(12) *péc-t*$_{\mathrm{V}}$ ‘to bake’ $\xrightarrow{\text{é>e, c>č}}$ *peč-en-í*$_{\mathrm{N}}$ ‘baking’

(13) *ovc-e*$_{\mathrm{N}}$ ‘sheep’ $\xrightarrow{\text{c>č}}$ *ovč-í*$_{\mathrm{N}}$ ‘belonging to/got from sheep’

(14) *vrat-a*$_{\mathrm{N}}$ ‘gate’ $\xrightarrow{\text{a>á}}$ *vrát-ný*$_{\mathrm{N}}$ ‘porter’

(15) *vrát-i-t*$_{\mathrm{V}}$ ‘to return’ $\xrightarrow{\text{á>a}}$ *vrat-ný*$_{\mathrm{A}}$ ‘returnable’

(16) *slav-ist-a*$_{\mathrm{N}}$ ‘Slavist’ → *slav-is-tický*$_{\mathrm{A}}$ ‘Slavic’

(17) *sláv-ist-a*$_{\mathrm{N}}$ ‘supporter of the sport club Slávie’ → *sláv-ist-ický*$_{\mathrm{A}}$ ‘belonging to the supporters of Slávie’

There are nearly 90 pairs of alternating graphemes in Czech. Since we model derivational relations as oriented from the base word to the derived one, the pairs of alternating graphemes are described as being oriented, too. The “base grapheme” (in the base) vs. the “target grapheme” (in the derivative) are discerned. Pairs of alternating graphemes differ in whether one of the them is always to be found as the base grapheme while the other one as the target grapheme across the lexicon, or if they are found in reverse order in other pairs of lexemes as well (so-called one-directional vs. bidirectional alternations, respectively, according to Osolsobě, 2002; Ziková, 2015, does not take orientation of the alternating graphemes into consideration). The *h*>*z* alternation in (18) is an example of the one-directional alternation in Czech. The graphemes *ch* and *š* enter the alternation *ch*>*š* on the one hand, and *š*>*ch* on the other ((19) vs. (20)).

(18) *drah-ý*$_{\mathrm{A}}$ ‘expensive’ $\xrightarrow{\text{h>z}}$ *draz-e*$_{\mathrm{D}}$ ‘at a high price’

(19) *tich-ý*$_{\mathrm{A}}$ ‘silent’ $\xrightarrow{\text{ch>š}}$ *tiš-e*$_{\mathrm{D}}$ ‘silently’

(20) *po-těš-i-t*$_{\mathrm{V}}$ ‘to please’ $\xrightarrow{\text{š>ch}}$ *po-těch-a*$_{\mathrm{N}}$ ‘pleasure’

The following classification differentiates five types of vowel alternations (A to E), three types of consonant alternations (F to H), and a type of mixed alternations (I; Dokulil, 1962, pp. 162ff, Osolsobě, 2002). The vowel (i.e. vowel-to-vowel) alternations are classified according to the quantity and quality of the base and target graphemes:

A) in quantitative alternations, a vowel is substituted for the same vowel with opposite quantity (short vowels are lengthened (21), long vowels shortened (22)):

   (21) *vy-jet*$_{\mathrm{V}}$ ‘to leave’ $\xrightarrow{\text{y>ý}}$ *vý-jezd*$_{\mathrm{N}}$ ‘leaving’

   (22) *tráv-a*$_{\mathrm{N}}$ ‘grass’ $\xrightarrow{\text{á>a}}$ *trav-natý*$_{\mathrm{A}}$ ‘grassy’

B) in qualitative alternations, a vowel is replaced by a different vowel with the same quantity:

(23) *hrab-a-t*$_{V}$ 'to dig' $\xrightarrow{a>o}$ *hrob*$_{N}$ 'grave'

C) in quantitative-qualitative alternations, a vowel in the base word is replaced by a qualitatively different vowel with opposite quantity in the target word:

(24) *říd-i-t*$_{V}$ 'to direct' $\xrightarrow{í>e}$ *řed-i-tel*$_{N}$ 'director'

(25) *ostrov*$_{N}$ 'island' $\xrightarrow{o>ů}$ *ostrův-ek*$_{N}$ (dimin.)

D) vowel deletion can be described as a type of vowel alternations, too; a vowel (mostly *e* in Czech derivation) is substituted by a zero (vowel-zero alternation):

(26) *pes*$_{N}$ 'dog' $\xrightarrow{e>0}$ *ps-í*$_{A}$ 'belonging to dog'

(27) *such-ý*$_{A}$ 'dry' $\xrightarrow{u>0}$ *sch-nou-t*$_{V}$ 'to become dry'

E) vowel insertion is described as a replacement of a zero by a vowel (zero-vowel alternation):

(28) *hr-á-t*$_{V}$ 'to play' $\xrightarrow{0>e}$ *her-n-a*$_{N}$ 'playroom'

The following types of consonant-to-consonant alternations are applied in Czech:

F) individual alternations when a single consonant is substituted by another one:

(29) *čern-ý*$_{A}$ 'black' $\xrightarrow{n>ň}$ *čerň*$_{N}$ 'black (colour)'

(30) *čern-och*$_{N}$ 'black man' $\xrightarrow{ch>š}$ *čern-oš-ka*$_{N}$ 'black woman'

G) consonant deletion and insertion is peripheral in contemporary Czech, cf. deletions in verb-to-verb derivation (31) and in derivation from proper nouns of foreign origin (32), and insertion of the initial *j* (which is not a prefix) in (33) :

(31) *top-i-t*$_{V}$ 'to drawn'' $\xrightarrow{p>0}$ *to-nou-t*$_{V}$ 'to be drawning'

(32) *Hamburk*$_{N}$ 'Hamburg' $\xrightarrow{k>0}$ *hambur-ský*$_{A}$ 'from Hamburg'

(33) *mí-t*$_{V}$ 'to have' $\xrightarrow{0>j}$ *jmě-ní*$_{N}$ 'property'

H) a substitution of a pair of consonants by a particular pair of consonants is called a group alternation:

(34) *měst-ský*$_{A}$ 'urban' $\xrightarrow{st>šť}$ *měšť-an*$_{N}$ 'burgher'

(35) *čes-ký*$_{A}$ 'Czech' $\xrightarrow{sk>št}$ *češ-tina*$_{N}$ 'Czech language'

I) In addition, in so-called mixed alternations, a vowel is replaced by a combination of a vowel and constant; this type is mostly found in deverbal derivation:

(36) *stá-t*$_{V}$ 'to stand' $\xrightarrow{á>oj}$ *stoj-ící*$_{A}$ 'standing'

(37) *stá-t*$_{V}$ 'to stand' $\xrightarrow{\text{á>av}}$ *po-stav-i-t*$_{V}$ 'to set up'

(38) *bí-t* $_{V}$ 'to beat' $\xrightarrow{\text{í>ij}}$ *bij-ící*$_{A}$ 'beating'

An alternative classification (into vowel-zero alternations, quantitative alternations, and palatalization alternations) was proposed by Ziková (2015).

3.2. Distribution of morphographemic alternations

In Czech derivation, alternations affect almost the entire repertory of graphemes and all types of morphemes (and, assumably, a considerable part of the Czech lexicon).[12] Considering the repertory of graphemes in Czech, all vowels and consonants, except for *p*, *b*, *f*, *v*, *m*, and *l*, enter alternations. Both vowel and consonant alternations can occur at any position in a word, even at the first one ((39) to (41)).

(39) *úz-ký*$_{A}$ 'narrow' $\xrightarrow{\text{ú>u}}$ *uz-oučký*$_{A}$ 'very narrow'

(40) *hn-á-t*$_{V}$ 'to drive' $\xrightarrow{\text{h>ž, 0>e}}$ *žen-oucí*$_{A}$ 'driving'

(41) *hr-á-t*$_{V}$ 'to play' $\xrightarrow{\text{r>ř}}$ *hř-iště*$_{N}$ 'playground'

Individual alternating pairs differ in frequency.[13] According to an overall estimate provided by Osolsobě (2002), *a* and *á* out of the vowels enter alternations most frequently (*a* changes into *á*, *e*, *ě*, and *o*, the long *á* into *a*, *e*, *i*, and *í*). The pairs *s*>*š*, *k*>*c*, and *c*>*č* are the most frequent consonant alternations. The vowel *o* and the consonant *g* alternate least frequently. Nevertheless, neither the quality nor the frequency of alternating graphemes allow for estimating the productivity of particular alternations (cf. Ziková, 2016a).

Alternations affect all types of morphemes, namely prefixes, roots, and suffixes during derivation (and roots and suffixes during inflection, see the next subsection). Vowel lengthening (plus the alternations *o*>*ů*) occurs in prefixes, roots as well as suffixes, whereas other vowel alternations (shortening, qualitative alternations, and

[12]The amount of words affected by alternations was preliminarily estimated in our study including 500 nouns, adjectives, verbs, and adverbs (consisting of at least two characters, only the first of which was allowed to be uppercased) with the highest token frequency in the representative corpus of Czech (SYN2015, 120 million tokens; Křen et al., 2015). 100 (20 %) out of the examined lemmas involved alternations with respect to their particular base words. For 271 (54.2 %) out of 500 lemmas, it was possible to find at least one derived word that was affected by alternations.
The aim of the study was not to estimate the alternation frequency in the overall data collection. As "phonetic change often progresses often more quickly in items with high token frequency" (Bybee, 2001, p. 11; cf. also Bybee, 2007, p. 270), less alternations are expected in words with lower frequency. We believe that a more precise picture of how alternations are distributed over the lexicon could be inferred from the DeriNet data.

[13]Here and elsewhere in the paper, type frequency in the Czech lexicon is meant if we do not refer to a particular corpus or another data resource.

quantitative-qualitative alternations) are limited to roots and suffixes. Mixed alternations are limited to derivation from verbs and affect final vowels of the root morpheme (these alternations originate in inflection; see Sect. 4.3). Alternations with zero (in both directions) are prototypically found in roots, or less frequently, in suffixes. Consonants alternate mostly in the final position of the stem, forced by the added suffix. Group alternations affect either two final consonants of the stem, or the final consonant of the stem and the first one of the suffix.

In Appendix, we provide an exhaustive list of alternations as observed in the lexicon of contemporary Czech, specifically as manifested in writing. Neither the origin of the alternation,[14] nor the frequency or productivity in the lexicon were taken into consideration. Nearly 90 alternation pairs are listed in alphabetical order according to the form of the alternating grapheme in the base word. Each pair is given in a separate line, the direction of the alternation is of significance. If a pair of graphemes alternates in both directions, it is listed twice in the list (indicated with the note "bidir." with each of the directions). Each pair of alternating graphemes is followed by a set of examples with the particular alternation in prefix, root and suffix (if available). In the rightmost column, we tried to find counter-examples, documenting that a particular grapheme even in a close morphosyntactic context does not necessarily undergo the same change.

3.3. Alternations in derivation vs. in inflection

Most of the morphographemic alternations are found in both derivation and inflection.[15] There are only few pairs limited either to the former, or to the latter area; e.g. the *é*>*í* and *é*>*ý* alternations are found in derivation only (42), the *g*>*z* alternation exclusively in inflection (43). Apart from the distribution (alternations in inflection do not occur in prefixes), the alternations exhibit the same features in inflection as in derivation, esp. massive presence and irregularity.

(42) *polévka*$_{\mathrm{N}}$ 'soup' $\xrightarrow{\text{é>í}}$ *polívka*$_{\mathrm{N}}$ 'soup'

(43) *filolo***g**$_{\mathrm{N}}$ 'philologist', *filolo***z**-*ích*$_{\mathrm{loc.sg.masc.anim}}$

Dokulil (1962, p. 112) pointed out the complicated relations between alternations in a particular word and in a word derived from it. He examined the inflectional

[14] We thus omit the difference (pointed out by Dokulil, 1962, pp. 11f) between alternations that are required by a certain word-formation type (they depend on the graphemic structure of the affix and are obligatory, or accompany a certain word-formation type) and alternations that are not – from the synchronic point of view – related to the particular word-formation type "and are thus not considered word-formation alternations" [translated by the author of the paper] (in spite of being systematic in diachrony; e.g. *a*>*ě* in *svatý* 'holy' → *světec* 'holy man').

[15] The question to which linguistic subdiscipline alternations belong to has been discussed across different approaches (see Bybee and Brewer, 1980, or Bermúdez-Otero and McMahon, 2006, for summaries).

paradigms of both the base and the derivative whether they share an alternation. Here is a simplified list of types based on Dokulil's findings:

1. the derivative (its lemma and all inflected forms) exhibits an alternation with respect to the lemma and all inflected forms of the base word (the particular alternation is not present in the inflectional paradigm of the base word):

   (44) inflection of the base word: *čáp*$_{\mathrm{N}}$ 'stork', *čáp-a*$_{\mathrm{gen.sg}}$, *čáp-ovi*$_{\mathrm{dat.sg}}$ etc.
   derivation: *čáp*$_{\mathrm{N}}$ 'stork' $\xrightarrow{\text{á>a}}$ *čap-í*$_{\mathrm{A}}$ 'belonging to stork' (inflection of the derived word: *čap-ího*$_{\mathrm{gen.sg}}$, *čap-ímu*$_{\mathrm{dat.sg}}$ etc.)

   (45) infl. of the base word: *sprav-ova-t*$_{\mathrm{V}}$ 'administrate', *sprav-uj-i*$_{\mathrm{1.sg.pres.act}}$ etc.
   derivation: *sprav-ova-t*$_{\mathrm{V}}$ 'administrate' $\xrightarrow{\text{a>á}}$ *správ-a*$_{\mathrm{N}}$ 'administration' (inflection of the derived word: *správ-y*$_{\mathrm{gen.sg}}$, *správ-ě*$_{\mathrm{dat.sg}}$ etc.)

2. the derived word exhibits an alternation in its entire inflectional paradigm with respect to the lemma of the base word; however, the alternation is involved in some inflectional forms of the base word:

   (46) inflection of the base word: *d***ů***m*$_{\mathrm{N}}$ 'house', *d***o***m-u*$_{\mathrm{gen.sg}}$, *d***o***m-u*$_{\mathrm{dat.sg}}$, *dům*$_{\mathrm{acc.sg}}$ etc.
   derivation: *dům*$_{\mathrm{N}}$ 'house' $\xrightarrow{\text{ů>o}}$ *dom-ek*$_{\mathrm{N}}$ (dimin.) (inflection of the derived word: *dom-k-u*$_{\mathrm{gen.sg}}$, *dom-k-u*$_{\mathrm{dat.sg}}$, *dom-ek*$_{\mathrm{acc.sg}}$ etc.)

   (47) inflection of the base word: *b***ůh**$_{\mathrm{N}}$ 'god', *b***o***h-a*$_{\mathrm{gen.sg}}$, *b***o***h-u*$_{\mathrm{dat.sg}}$, *b***o***h-a*$_{\mathrm{acc.sg}}$, *b***o***že*$_{\mathrm{voc.sg}}$, *b***o***z-i*$_{\mathrm{nom.pl}}$ etc.
   derivation: *bůh*$_{\mathrm{N}}$ 'god' $\xrightarrow{\text{h>ž}}$ *bůž-ek*$_{\mathrm{N}}$ (dimin.) (inflection of the derived word: *bůž-k-a*$_{\mathrm{gen.sg}}$, *bůž-k-ovi*$_{\mathrm{dat.sg}}$ etc.)
   derivation: *bůh*$_{\mathrm{N}}$ 'god' $\xrightarrow{\text{ů>o, h>ž}}$ *bož-í*$_{\mathrm{A}}$ 'god's' (inflection of the derived word: *bož-ího*$_{\mathrm{gen.sg}}$, *bož-ímu*$_{\mathrm{dat.sg}}$ etc.)

3. the alternation that exhibits the lemma of the derivative with respect to the lemma of the base occurs in the inflected forms of the base (cf. type 1) but, moreover, inflectional forms of the derivative include an alternation with respect to the lemma of the derivative but not to the lemma of the base:

   (48) inflection of the base word: *sta***r***-ý*$_{\mathrm{A}}$ 'old', *star-ého*$_{\mathrm{gen.sg.masc.anim}}$, *sta***ř***-í*$_{\mathrm{nom.pl.masc.anim}}$ etc.
   derivation: *starý*$_{\mathrm{A}}$ 'old' $\xrightarrow{\text{r>ř}}$ *stař-ec*$_{\mathrm{N}}$ 'old man' (inflection of the derived word: *sta***r***-c-e*$_{\mathrm{gen.sg}}$, *sta***r***-c-i*$_{\mathrm{dat.sg}}$ etc.)

The fact that the alternation observed between a base lemma and the lemma of the derivative can be found in inflectional forms of the base (as in the type 2 and 3) were employed in order to find base words for words with alternations in the root

morpheme in DeriNet (see Sect. 4.4). The type 1 above and the relations between the inflectional forms of the derived word and of the base in 3 were not relevant for our purpose.

4. Alternations in the DeriNet database of derivational relations

In this section, the methods used for the establishment of derivational links in DeriNet are described; the main focus is on which type of morphographemic alternations was modelled by the individual method (for general aspects of the build-up of the database see Ševčíková and Žabokrtský, 2014b; Žabokrtský et al., 2016). String-substitution rules, which constitute the methodological core of our approach (Sect. 4.1 and 4.2), were efficient for modelling frequent alternations in the final grapheme of the stem. A significant portion of derivational relations, often with multiple alternations connected with deverbal derivation, was extracted from the inflectional dictionary MorfFlex CZ (Sect. 4.3). In order to cover alternations in roots that emerged for small groups of words or even for individual words only, inflectional paradigms were exploited for alternations and used for the search of the base-target pairs in DeriNet (Sect. 4.4). Alternations connected with prefixation of verbs were handled separately (Sect. 4.5).

4.1. Searching base adjectives for selected groups of derived words

The DeriNet database was initialised in 2013 to underpin the linguistic research project on deadjectival derivation in Czech with a solid data resource. In a set of lexemes extracted from a large corpus of Czech (Bojar et al., 2012), base adjectives were searched for selected groups of derived words. Deadjectival nouns and adverbs were linked to the base adjectives using heuristics that were manually compiled as regular expressions substituting the final string of the derived word for an adjectival string.[16] For instance, the derivational rule in (49), based on the respective regular expression, was used to identify pairs of an adjective (A) ending in *-ý* and a noun (N) consisting of the same grapheme string except for the final *-ost* instead of the adjectival *-ý*. Only few analogous rules were sufficient to cover all nouns in *-ost* (cf. (50)) and to link most of the adverbs with their base adjectives (51).

(49) A-ý>N-ost: *závislý*$_{\mathrm{A}}$ ‘dependent’ → *závislost*$_{\mathrm{N}}$ ‘dependency’

(50) A-í>N-ost: *revoluční*$_{\mathrm{A}}$ ‘revolutionary’ → *revolučnost*$_{\mathrm{N}}$ ‘revolutionarity’
A-í>N-nost: *budoucí*$_{\mathrm{A}}$ ‘future’ → *budoucnost*$_{\mathrm{N}}$ ‘future’

(51) A-ý>D-e: *bílý*$_{\mathrm{A}}$ ‘white’ → *bíle*$_{\mathrm{D}}$ ‘white(ly)’
A-ý>D-ě: *krutý*$_{\mathrm{A}}$ ‘cruel’ → *krutě*$_{\mathrm{D}}$ ‘cruelly’

[16] The strings corresponded either to suffixes, or to inflectional endings, or were longer (and included one or even more characters of the root morpheme). In the paper, rules based on these strings are therefore generally called “string-substitution rules”.

A-í>D-ě: *revoluční*$_{\text{A}}$ 'revolutionary' → *revolučně*$_{\text{D}}$ 'revolutionary'
A-ý>D-y: *přátelský*$_{\text{A}}$ 'friendly' → *přátelsky*$_{\text{D}}$ 'in a friendly way'

At this phase, alternations did not seem to be a significant issue since they were not frequent in our sample of deadjectival derivation or, more precisely, many of the alternations were not mirrored in writing. For instance, in the most frequent group of deadjectival adverbs (with the suffix *-ě*), or in adjectives in *-ičký* and *-inký*, the final consonant of the root is palatalized in pronunciation but stays unpalatalized in writing as the palatalization is represented by the initial vowel of the suffix (52), (53).

(52) *pěkný*$_{\text{A}}$ 'nice' → *pěkně*$_{\text{D}}$ 'nicely'

(53) *chudý*$_{\text{A}}$ 'poor' → *chudičký*$_{\text{A}}$ 'dirt-poor'

There were only several hundreds of derived words with alternations in the data set in total. Alternations in the final graphemes of the stem were encoded in specific derivational rules such as (54) and (55). The entire word-formation type of deadjectival names of languages in *-ina* which includes a group alternation (*sk*>*št* or *ck*>*čt*) was possible to be covered only by two rules in (56).

(54) A-cký>N-čnost: *praktický*$_{\text{A}}$ 'practical' → *praktičnost*$_{\text{N}}$ 'practicality'

(55) A-ký>D-ce: *blízký*$_{\text{A}}$ 'close' → *blízce*$_{\text{D}}$ 'closely'
A-chý>D-še: *jednoduchý*$_{\text{A}}$ 'simple' → *jednoduše*$_{\text{D}}$ 'simply'
A-rý>D-ře: *dobrý*$_{\text{A}}$ 'good' → *dobře*$_{\text{D}}$ 'well'

(56) A-ský>N-ština: *arabský*$_{\text{A}}$ 'Arabic' → *arabština*$_{\text{N}}$ 'Arabic language'
A-cký>N-čtina: *anglický*$_{\text{A}}$ 'English' → *angličtina*$_{\text{N}}$ 'English language'

In our data sample, only few base adjectives underwent alternations in the root. However, since there was a varied spectrum of vowel alternations in the roots and they occurred selectively with individual affixes (see (57) and (58)), base-target pairs with root alternations were identified individually and linked manually in the data.

(57) *mladý*$_{\text{A}}$ 'young' $\xrightarrow{\text{a>á}}$ *mládí*$_{\text{N}}$ 'youth'
mladý$_{\text{A}}$ 'young' $\xrightarrow{\text{a>á}}$ *mládě*$_{\text{N}}$ 'baby animal'
mladý$_{\text{A}}$ 'young' → *mladě*$_{\text{D}}$ 'in a young manner'
mladý$_{\text{A}}$ 'young' $\xrightarrow{\text{a>á}}$ *mládnout*$_{\text{V}}$ 'to become younger'

(58) *bílý*$_{\text{A}}$ 'white' $\xrightarrow{\text{í>ě}}$ *běloučký*$_{\text{A}}$ 'purely white'
bílý$_{\text{A}}$ 'white' $\xrightarrow{\text{í>ě}}$ *běloušN* 'white horse'
bílý$_{\text{A}}$ 'white' $\xrightarrow{\text{í>ě}}$ *bělit*$_{\text{V}}$ 'to bleach'
bílý$_{\text{A}}$ 'white' → *bílit*$_{\text{V}}$ 'to paint white'

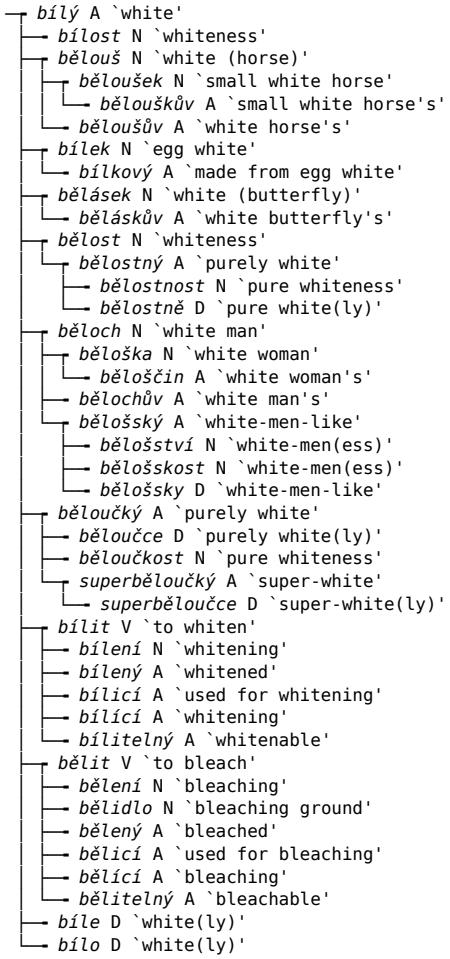


Figure 1. Derivational tree with the root adjective bílý

A derivational tree with the root adjective *bílý* 'white' is displayed in Fig. 1.[17] In the tree, each lexeme is connected by an edge with its direct base word; each edge thus corresponds to a single derivational step. There are five nouns derived directly from *bílý*, namely *bílost*, *běloušˇ*, *bílek*, *bělásek*, *bělost*, and *běloch*, as listed from the top to the bottom of the tree. The noun *běloch* is the base word for the noun *běloška* and for two adjectives (*bělochův*, *bělošský*); on the latter adjective, three other lexemes (*bělošskost*, *bělošství*, *bělošsky*) are based (see the tree for English equivalents).

The adjectival data, under the title AdjDeriNet, were published in 2014 (consisting of app. 18 thousand adjectives with more than 26 thousand nouns, adjectives, verbs and adverbs derived from them; Ševčíková and Žabokrtský, 2014a).

4.2. Alternations in string-substitution rules

The decision to overcome the limitation to deadjectival derivation and to extend the repertory of derivational relations (with the ambition of identifying as many derivational relations in the data as possible) was connected with an attempt to automatize the process of identification of the candidate base-target pairs.

Based on the assumption that lemmas that share a sufficiently long sequence of characters are likely to be derivationally related, pairs of lemmas with a high string similarity (from the left-most character) were identified automatically and, subsequently, grouped according to the strings in which the pair members differed. The differing suffix strings were formalized as string-substitution rules for app. 400 most frequent groups. Out of the list of rules thus obtained, 35 rules were manually selected that reliably corresponded to derivational relations of a base word and a word immediately derived from it.[18] As the next step, the direction of the relation was determined in the rules. The string of the base word was mostly shorter than the corresponding string in the target word; see (59). A single rule often matched several word-formation types; e.g. the first rule in (59) covers derivation of feminine profession nouns from masculine counterparts as well as diminutivization of feminine nouns, the third rule deverbal derivation of both agentive nouns and instrument nouns. Only three out of 35 rules involved alternations, namely an alternation in the final consonant of the stem (60).

[17] In the paper, a simple tree representation was preferred to the graphical output provided by the tools DeriNet Search and DeriNet Viewer since the simple tree requires less space. The tools can be used online for searching the DeriNet data; see `http://ufal.mff.cuni.cz/derinet/search` and `http://ufal.mff.cuni.cz/derinet/viewer`.

[18] The remaining rules either matched less frequent relations, or corresponded to relations between pairs of words that are related only indirectly. For instance, the candidate rule A-ův>A-čin corresponds to the relation between a masculine and a feminine possessive adjective (*manželův* 'husband's' and *manželčin* 'wife's') that both belong into the same derivational family and, thus, into the same derivational tree but are not in the direct base-target relation.

(59) N-a>N-ka: *policista*$_{\mathrm{N}}$ 'policeman' → *policistka*$_{\mathrm{N}}$ 'policewoman', *škola*$_{\mathrm{N}}$ 'school' → *školka*$_{\mathrm{N}}$ 'kindergarden'
A-ý>N-ec: *báz-liv-ý*$_{\mathrm{A}}$ 'timid' → *báz-liv-ec*$_{\mathrm{N}}$ 'coward'
V-t>N-č: *bav-i-t*$_{\mathrm{V}}$ 'entertain' → *bav-i-č*$_{\mathrm{N}}$ 'entertainer', *vypín-a-t*$_{\mathrm{V}}$ 'to switch off' → *vypín-a-č*$_{\mathrm{N}}$ 'switch'

(60) N-ce>A-ční: *akc-e*$_{\mathrm{N}}$ 'action' → *akč-ní*$_{\mathrm{A}}$ 'action'
N-k>N-ček: *zob-ák*$_{\mathrm{N}}$ 'beak' → *zob-áč-ek*$_{\mathrm{N}}$ (dimin.)
N-ce>N-čka: *had-ic-e*$_{\mathrm{N}}$ 'hose' → *had-ič-ka*$_{\mathrm{N}}$ (dimin.)

The set of 35 rules was applied to the data in order to find candidate pairs of base-target words. Incorrect candidate pairs were excluded manually; e.g. in (61) both the suggested base and target nouns are directly derived from the verb *rýt* 'to engrave' (in examples, the correct base word follows in parentheses). A significant portion of the excluded pairs was defective due to the inappropriate approach to alternations (62).

(61) *rytý*$_{\mathrm{A}}$ 'engraved' $\not\rightarrow$ *rytec*$_{\mathrm{N}}$ 'engraver' (*rýt*$_{\mathrm{V}}$ 'to engrave' $\xrightarrow{\text{ý>y}}$ *rytec*$_{\mathrm{N}}$)

(62) *kareta*$_{\mathrm{N}}$ 'Caretta turtle' $\not\rightarrow$ *karetní*$_{\mathrm{A}}$ 'card (game)' (*karta*$_{\mathrm{N}}$ 'card' $\xrightarrow{\text{0>e}}$ *karetní*$_{\mathrm{A}}$)
let$_{\mathrm{N}}$ 'flight' $\not\rightarrow$ *letní*$_{\mathrm{A}}$ 'summer (time)' (*léto*$_{\mathrm{N}}$ 'summer' $\xrightarrow{\text{é>e}}$ *letní*$_{\mathrm{A}}$)

A significantly larger list of (app. 450) string-substitution rules was compiled manually from a representative grammar of Czech (Karlík et al., 2000). As compared to the automatically extracted rules used in the previous step, the manually compiled rules concerned less frequent word-formation types (for instance, deverbal nouns denoting actions, or collective nouns derived from nouns (63)) and, moreover, some of them included highly frequent morphographemic alternations of all types (vowel insertion and deletion, quantitative vowel alternations, and palatalization). Thus, for instance, the rules in (64) match derivation of feminine profession nouns from masculines without an alternation and with the alternation *c*>*č*, which is very frequent with this word-formation type. Similarly, the second rule in (65) includes a frequent vowel deletion that is associated with the derivation of adjectives from nouns. The application of the rules on the data was followed by manual annotation of incorrect base-target pairs, similarly as with the automatically extracted rules.

(63) V-it>N-ba: *léčit*$_{\mathrm{V}}$ 'to treat' → *léčba*$_{\mathrm{N}}$ 'treatment'
N->N-stvo: *člen*$_{\mathrm{N}}$ 'member' → *členstvo*$_{\mathrm{N}}$ 'members'

(64) N->N-ka: *učitel*$_{\mathrm{N}}$ 'teacher' → *učitelka*$_{\mathrm{N}}$ 'female teacher'
N-c>N-čka: *herec*$_{\mathrm{N}}$ 'actor' → *herečka*$_{\mathrm{N}}$ 'actress'

(65) N->A-ový: *achát*$_{\mathrm{N}}$ 'agate' → *achátový*$_{\mathrm{A}}$ 'agate'
N-ek>A-kový: *bílek*$_{\mathrm{N}}$ 'egg white' → *bílkový*$_{\mathrm{A}}$ 'made from egg white' (see Fig. 1)

Table 1. Frequency list of 28 alternation pairs applied in the string-substitution rules

altern.	freq.	altern.	freq.	altern.	freq.	altern.	freq.	altern.	freq.
i>*í*	594	*á*>*a*	104	*h*>*ž*	31	*n*>*ň*	13	*r*>*ř*	4
í>*i*	314	*ě*>*í*	77	*o*>*u*	26	*u*>*ou*	9	*é*>*í*	2
í>*ě*	191	*z*>*ž*	53	*ý*>*y*	21	*o*>*ů*	8	*e*>*é*	2
ou>*u*	178	*a*>*á*	50	*é*>*e*	21	*d*>*ď*	8	*u*>*ú*	1
s>*š*	116	*y*>*ý*	36	*ch*>*š*	17	*ů*>*o*	7		
e>*í*	114	*k*>*č*	36	*c*>*č*	15	*t*>*ť*	7		

Since so far we were able to cover only alterations explicitly encoded in the string-substitution rules the amount of which was still rather limited, an experiment was carried out that allowed alternations in stems during application of both automatically extracted and manually compiled rules. A total of 18 vowel alternations and 10 consonant alternations were selected in advance and applied mechanically together with each string-substitution rule. No more than one alternation was allowed in each pair in order to prevent unmanageable overgeneration of base-target candidate pairs. Nevertheless, if the alternation was applied together with a string-substitution rule that encoded an alternation too, derivations with up to two alternations might be covered for the first time (66).

(66) N-k>A-čí: *ptákN* ‘bird’ $\xrightarrow{\text{á>a, k>č}}$ *ptačíA* ‘bird’s’

Examples of incorrectly suggested pairs were rejected by a human annotator (67). In total, alternations were applied with more than 1,600 derivational links confirmed within the manual annotation; see the frequency list in Table 1.

(67) *kuřeN* ‘chicken’ $\xrightarrow{\text{u>ou}}$ *kouřovýA* ‘smoky’ (*kouřN* ‘smoke’ → *kouřovýA*)
ženaN ‘woman’ $\xrightarrow{\text{e>í}}$ *žíněnýA* ‘made of horsehair’ (*žíněN* ‘horsehair’ → *žíněnýA*)
cenaN ‘price’ $\xrightarrow{\text{e>í}}$ *cínovýA* ‘made of tin’ (*cínN* ‘tin’ → *cínovýA*)
celaN ‘cell’ $\xrightarrow{\text{e>í}}$ *cílovýA* ‘finishing’ (*cílN* ‘finish’ → *cílovýA*)

Application of string-substitution rules was extremely efficient. The steps described in this subsection yielded 350 thousand derivational relations in DeriNet 1.4.

4.3. Alternations extracted from the MorfFlex CZ dictionary

Another considerable portion of derivational links in the DeriNet database was extracted from the MorfFlex CZ dictionary. In MorfFlex CZ, derivational information was encoded as a part of the so-called technical suffix of the lemma (Hajič, 2004;

Hana et al., 2005). The technical suffix *2*t* in (68) means that by substituting two final graphemes of the lemma for the grapheme *t*, the base of the adjective is reconstructed (the verb *hubnout* 'to lose weight').

(68) *hubnoucí_^(*2t)* 'losing weight'

MorfFlex CZ was exploited to identify base words for high-frequency groups of words derived mostly from verbs. Derivation from verbs in Czech is specific in that inflected verbal forms rather than the infinitive itself often serve as the base word in derivation. However, since individual verb forms are not involved in DeriNet and are all represented by the infinitive, words derived from different verbal forms had to be linked up to the particular infinitive in the database. Radical changes in the formal shape of deverbal adjectives with respect to the base infinitive are demonstrated in (69) and (70), the changes though do not in fact relate to derivation, but are to be traced back to the inflection (the respective verbal form that entered the derivation is given in square brackets after the infinitive). The formation of deverbal adjectives from transgressives (69) and participles (70) is thus very close to inflection and has been discussed as a transition zone between inflection and derivation in Czech (Dokulil, 1962, pp. 44; Karlík et al., 2000, pp. 172f).

(69) *hnát*$_{\mathrm{V}}$ 'to drive' [*ženouc*$_{\mathrm{transgr.fem.sg.pres.act.impf}}$] → *ženoucí*$_{\mathrm{A}}$ 'driving'
setřít$_{\mathrm{V}}$ 'to wipe' [*setřevši*$_{\mathrm{transgr.fem.sg.past.act.pf}}$] → *setřevší*$_{\mathrm{A}}$ 'wiped'

(70) *hnát*$_{\mathrm{V}}$ 'to drive' [*hnán*$_{\mathrm{3.sg.masc.ptcp.pass.impf}}$] → *hnaný*$_{\mathrm{A}}$ 'driven'
projít$_{\mathrm{V}}$ 'to expire' [*prošel*$_{\mathrm{3.sg.masc.ptcp.past.pf}}$] → *prošlý*$_{\mathrm{A}}$ 'expired'

In addition to the deverbal derivation, the technical suffixes were related to possessive adjectives derived from nouns, which are another word-formation type from the transition zone between derivation and inflection. Some of the most frequent technical suffixes exploited in DeriNet are listed in Table 2. For each suffix, an example lemma is provided and the suffix information is reformulated as a base-target pair.

As for morphographemic alternations, the technical suffixes *3*at* and *3*it* are examples of vowel alternations in the final grapheme of the stem; vowel deletion is encoded in the last technical suffix in Table 2.

Derivational information from the technical suffixes in MorfFlex CZ was used to establish 399 thousand derivational links in DeriNet 1.4.

4.4. Exploiting inflectional paradigms for description of alternations in derivation

Although alternations were included in each of the methods reported on so far, we were still not satisfied with the coverage of lexemes with alternations. Therefore, a method was proposed that targeted specifically at connecting derived words containing alternations with the correct base word. It focused on alternations that were difficult to cover with the methods described above, especially on changes occurring

Table 2. Technical suffixes of lemmas in MorfFlex used for the establishment of derivational links in DeriNet

technical suffix	lemma with the technical suffix	corresponding derivational relation
**2t*	*hubnoucí_^(*2t)*	*hubnout*$_{\mathrm{V}}$ 'to slim down' → *hubnoucí*$_{\mathrm{A}}$ 'slimming'
	*marinovaný_^(*2t)*	*marinovat*$_{\mathrm{V}}$ 'marinate' → *marinovaný*$_{\mathrm{A}}$ 'marinated'
**4*	*popsatelný_^(*4)*	*popsat*$_{\mathrm{V}}$ 'to describe' → *popsatelný*$_{\mathrm{A}}$ 'describable'
**3at*	*dělání_^(*3at)*	*dělat*$_{\mathrm{V}}$ 'to do' → *dělání*$_{\mathrm{N}}$ 'doing'
**3it*	*bílení_^(*3it)*	*bílit*$_{\mathrm{V}}$ 'to whiten' → *bílení*$_{\mathrm{N}}$ 'whitening' (see Fig. 1)
**2*	*manželův_^(*2)*	*manžel*$_{\mathrm{N}}$ 'husband' → *manželův*$_{\mathrm{A}}$ 'husband's'
**3ec*	*otcův_^(*3ec)*	*otec*$_{\mathrm{N}}$ 'father' → *otcův*$_{\mathrm{A}}$ 'father's'

"deeper" in the root morpheme and on vowel deletion in suffixes. We exploited the fact that alternations which are identified in derived words with respect to their base words might be identical with those observed in the inflectional paradigm of the respective base word (cf. an analogous approach for French by Bonami et al., 2009).

Since lemmas in DeriNet were taken from the inflectional dictionary MorfFlex CZ and both resources are interconnected, information on inflection of DeriNet lemmas is easily accessible. The core issue that alternations are not marked in MorfFlex CZ has been overcome by a provisional, rather technical solution. Each lemma was compared letter-wise from left to right against each of its inflected forms. At least one final grapheme of the inflected form was supposed to be the inflectional ending and thus was not included into the comparison. The inflectional string was marked as containing an alternation, if at any position the character in the lemma differed from that in the inflected form and the pair of differing characters was found in the list of 30 alternations pairs.[19] Due to the alternations *e*>0 and 0>*e*, the lemma might be longer then the inflected substring (71), or the other way round (72). Inflectional strings with one to three alternations with respect to the lemma ("alternated strings") were identified. For a single lemma, more formally different alternated strings could be listed (72).

(71)

lemma	*k*	*r*	*k*	*a*	*v*	***e***	***c***	*krkavec* 'raven'
alternated string	*k*	*r*	*k*	*a*	*v*	**0**	***č***	*krkavč*

(72)

lemma	*d*	*v*	*ů*	***r***	**0**	*dvůr* 'yard'
alternated string 1	*d*	*v*	*o*	*r*	0	*dvor*
alternated string 2	*d*	*v*	*o*	*r*	***e***	*dvore*
alternated string 3	*d*	*v*	*o*	***ř***	0	*dvoř*

[19]The list in Table 1 was enriched with the alternations *e*>0 and 0>*e* for this purpose.

The list of lemmas and corresponding alternated strings was used as input data for the string-substitution rules compiled in the previous steps of the annotation process. The string-substitution (esp. string-adding) rules were applied to the alternated strings instead of to the lemma and the existence of output string suggested by the rule was attested in the data. Manual annotation confirmed app. 2,300 derivational relations that have not been identified so far.

For instance, the rule in (73) was applied on the alternated string from (71) in order to create a link to a derived adjective; the first one out of the alternated strings in (72) turned out to be most effective for creating links between *dvůr* and its derivatives (74).

(73) N->A-í: *krkavec*$_{\mathrm{N}}$ 'raven' / *krkavč* → *krkavčí*$_{\mathrm{A}}$ 'belonging to raven'

(74) N->N-ec: *dvůr*$_{\mathrm{N}}$ 'yard'/ *dvor* → *dvorec*$_{\mathrm{N}}$ 'court'
N->N-ek: *dvůr*$_{\mathrm{N}}$ 'yard'/ *dvor* → *dvorek*$_{\mathrm{N}}$ (dimin.)
N->A-ský: *dvůr*$_{\mathrm{N}}$ 'yard'/ *dvor* → *dvorský*$_{\mathrm{A}}$ 'court (etiquette)'
N->A-ní: *dvůr*$_{\mathrm{N}}$ 'yard'/ *dvor* → *dvorní*$_{\mathrm{A}}$ 'court (lady)'

4.5. Alternations in derivation of verbs from verbs

Derivation of verbs from verbs was addressed separately. In Czech, verbs are derived from verbs by suffixation and prefixation. Prefixation is even prevailing in formation of verbs whereas suffixes predominate over prefixes in derivation of other part-of-speech categories in Czech. Deverbal derivation of verbs is connected with a significant amount of alternations.Deverbal prefixation and suffixation of verbs are both closely interconnected with the category of aspect.[20]

Prefixation either changes imperfective verbs into perfective ones (see the pure aspectual pair of verbs in (75)), or modifies the lexical meaning of an imperfective ((77) and (76)) or perfective verb (resulting in another perfective; (78)). In monosyllabic verbs with a long vowel, the vowel is shortened during prefixation systematically (in addition to (75) and (76), verbs *znát* 'to know', *brát* 'to take', *hnát* 'to ride' belong to this group). Suffixation is used especially to form imperfective counterparts from perfective verbs (79), to derive iterative verbs from imperfectives (80), or secondary imperfectives from prefixed perfectives (81).[21] Suffixation is connected mostly with

[20] In spite of a long-term discussion on the category of aspect, the status of this category is far from clear in Czech and other Slavic languages (e.g. Vey, 1952; Comrie, 1976; Mel'čuk, 1976; Kopečný, 1962; Komárek, 2006). In DeriNet, and thus in the present paper, derivation of verbs is treated with the primary focus on formal features, without respect to whether the affix changes just the aspect of the verb (e.g. "pure perfectivizing" prefixes in Czech) or whether it modifies the lexical meaning of the base verb. For a linguistically rooted discussion on the representation of derivational relations in verbal families with regard to the aspect see (Ševčíková et al., 2017, in press).

[21] The possibility to form a secondary imperfective is used to distinguish pure perfectivizing prefixes (cf. the prefixed derivative in ex. (75) from which the secondary imperfective cannot be derived) from other prefixes (cf. ex. (81) derived from the prefixed verb in ex. (76)).

Table 3. Prefixes used in the verb-to-verb derivation

ad-	*do-*	*na-*	*o-*	*ot-*	*pode-*	*při-*	*roze-*	*u-*	*vý-*	*zá-*
bez-	*dů-*	*ná-*	*ob-*	*ote-*	*pro-*	*před-*	*s-*	*ú-*	*vz-*	*ze-*
de-	*in-*	*nad-*	*obe-*	*pa-*	*přes-*	*půl-*	*se-*	*v-*	*vze-*	*zne-*
des-	*ko-*	*nade-*	*od-*	*po-*	*před-*	*re-*	*sou-*	*ve-*	*z-*	*zu-*
dez-	*kon-*	*ne-*	*ode-*	*pod-*	*předes-*	*roz-*	*sub-*	*vy-*	*za-*	*zů-*

alternations in the root morpheme that are often specific for the particular pair of verbs or are limited to small groups of verbs.

(75) *ps-á-t*$_{\texttt{Vimpf}}$ 'to write' $\xrightarrow{\text{á>a}}$ *na-ps-a-t*$_{\texttt{Vpf}}$ 'to write down'

(76) *ps-á-t*$_{\texttt{Vimpf}}$ 'to write' $\xrightarrow{\text{á>a}}$ *za-ps-a-t*$_{\texttt{Vpf}}$ 'to record'

(77) *skák-a-t*$_{\texttt{Vimpf}}$ 'to jump' $\rightarrow$ *vy-skák-a-t*$_{\texttt{Vpf}}$ 'to jump out'

(78) *skoč-i-t*$_{\texttt{Vpf}}$ 'to jump' $\rightarrow$ *vy-skoč-i-t*$_{\texttt{Vpf}}$ 'to jump out'

(79) *skoč-i-t*$_{\texttt{Vpf}}$ 'to jump' $\xrightarrow{\text{o>á, č>k}}$ *skák-a-t*$_{\texttt{Vimpf}}$ 'to jump'

(80) *skák-a-t*$_{\texttt{Vimpf}}$ 'to jump' $\rightarrow$ *skák-áva-t*$_{\texttt{Vimpf.iter}}$ 'to jump'

(81) *za-ps-a-t*$_{\texttt{Vpf}}$ 'to record' $\xrightarrow{\text{0>i}}$ *za-pis-ova-t*$_{\texttt{Vimpf}}$ 'to record'

As in the existing valency lexicon of Czech verbs Vallex (Lopatková et al., 2015) relations between aspectual pairs of verbs derived by suffixation are explicitly marked, we decided to extract these pairs as the first step in our task of creating derivational relations between verbs in DeriNet. Pairs of verbs that are not in a derivational relation (e.g. suppletive aspectual pairs such as *brát*$_{\texttt{Vimpf}}$ 'to take' – *vzít*$_{\texttt{Vpf}}$ 'to take') were excluded from the list. The usage of an existing, reliable lexical resource was preferred to the above presented methods (particularly string-substitution rules) precisely because of the heterogeneous nature of alternations in this type of suffixation. Second, a list of app. 50 prefixes used in deverbal derivation of verbs (vocalized variants listed as separate items; Table 3) was compiled and used to search Vallex for verbs that are derivationally related to the verbs in the extracted list of aspectual pairs. In these two steps, more than 3,100 verbs were found and preliminarily organized into 660 derivational families with a scope reaching from several tens of verbs (cf. families with the verbs *psát*$_{\texttt{Vimpf}}$ 'to write' or *skočit*$_{\texttt{Vpf}}$ 'to jump') up to pairs of verbs such as *šít*$_{\texttt{Vimpf}}$ 'to sew', *ušít*$_{\texttt{Vpf}}$ 'to finish sewing'.

The inner organization of the derivational families into a derivational tree consisting of oriented binary relations could not be inferred unambiguously from the data itself since there are complicated interconnections between the verbs with respect to

Figure 2. Alternative derivational trees with the root skočit *'to jump'. The derivational tree on the left is preferred in the presented approach (cf. ex. (77) to (80)).*

the form and aspectual characteristics that allow to organize the verbs in several competing ways. For instance, when modelling relations between the verbs *skočit*$_{\text{Vpf}}$ 'to jump', *skákat*$_{\text{Vimpf}}$ 'to jump', *vyskočit*$_{\text{Vpf}}$ 'to jump out', and *vyskákat*$_{\text{Vpf}}$ 'to jump out', the last verb can be interpreted either as the perfective counterpart of *vyskočit* formed through suffixation, or as a prefixed perfective derived from *skákat*. We preferred the prefixation (to creation of aspectual pairs by suffixation) to be a more important organizational principle in DeriNet, therefore, the latter interpretation was chosen in (77) and is mirrored in the tree structure on the left-hand side in Fig. 2; the former, refused interpretation corresponds to the tree on the right. The compared trees differ from the point of view of alternations; the preferred organization is connected with alternations along the single edge *skočit*$_{\text{Vpf}}$ 'to jump' → *skákat*$_{\text{Vimpf}}$ 'to jump'.

The following general guidelines for the inner organization of the derivational families into trees were specified:

- if an unprefixed aspectual pair is available in the derivational family (i.e. the aspectual pair differs in suffixes), the perfective verb is the root of the tree:
  e.g. *skočit*$_{\text{Vpf}}$ 'to jump' → *skákat*$_{\text{Vimpf}}$ 'to jump'
- if only an unprefixed imperfective is available with a prefixed perfective counterpart, the imperfective verb is the root of the tree:
  e.g. *šít*$_{\text{Vimpf}}$ 'to sew' → *ušít*$_{\text{Vpf}}$ 'to finish sewing'
- all prefixed perfectives are derived from the unprefixed counterpart;
  the counterpart is either perfective, e.g. *skočit*$_{\text{Vpf}}$ 'to jump' → *naskočit*$_{\text{Vpf}}$ 'to hop on' | *odskočit*$_{\text{Vpf}}$ 'to jump aside' | *poskočit*$_{\text{Vpf}}$ 'to jump up', ...,
  or imperfective, e.g. *skákat*$_{\text{Vimpf}}$ 'to jump' → *přeskákat*$_{\text{Vpf}}$ 'to jump over' | *vyskákat*$_{\text{Vpf}}$ 'to jump out' ...
- secondary imperfectives were linked to particular prefixed perfectives:
  e.g. *naskočit*$_{\text{Vpf}}$ 'to hop on' → *naskakovat*$_{\text{Vimpf}}$ 'to hop on', *poskočit*$_{\text{Vpf}}$ 'to jump up' → *poskakovat*$_{\text{Vimpf}}$ 'to jump up'
- iterative imperfectives as derived from the imperfective:
  e.g. *skákat*$_{\text{Vimpf}}$ 'to jump' → *skákávat*$_{\text{Vimpf.iter}}$ 'to jump' .

Having the derivational families organized into the tree structures (see Fig. 3), further verbs were searched for in the DeriNet data, using all pieces of information avail-

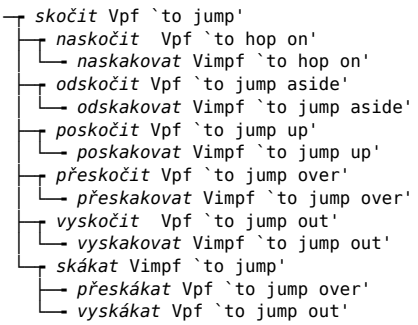


Figure 3. Derivational tree with the root verb skočit *'to jump' consisting of derivationally related verbs from the Vallex dictionary organized according to the adopted guidelines.*

able so far (in particular, string-substitution rules based on the tree structures and list of prefixes). The items were added into the derivational trees according to the guidelines. Compare the simplified derivational trees of the adjective *bílý*$_{\mathrm{A}}$ 'white' in Fig. 4; the simplification consists in displaying verbal nodes only (nodes of other part-of-speech categories were omitted).

The procedure described in Sect. 4.5 resulted in nearly 23 thousand new derivational relations in total. Since the newly connected verbs were mostly roots of subtrees consisting of direct and indirect deverbal derivatives, the new links led to connection of a number of trees into a structure with an extremely high number of nodes.[22]

5. Discussion and conclusions

In the paper, morphographemic alternations were approached from the perspective of semi-automatic modelling of derivational relations in the language resource specialized in derivational morphology of Czech. Methods of creating derivational links in DeriNet were presented with a focus on alternations covered by each of the methods. The method of exploiting inflectional paradigms developed specifically for dealing with alternations with respect to individual lexemes (Sect. 4.4) was less efficient (in terms of absolute frequency of created derivational links) than the string-substitution rules and derivational information from MorfFlex CZ, but it confirmed the feasibility and, also, usefulness of integrating inflectional information into description of derivation. As inflectional resources are elaborated for Czech more comprehensively than derivational data, which seems to be the case for other languages as well, the possible profits should be further explored.

[22]There are nearly 80 trees with more than 500 nodes each in DeriNet 1.4.

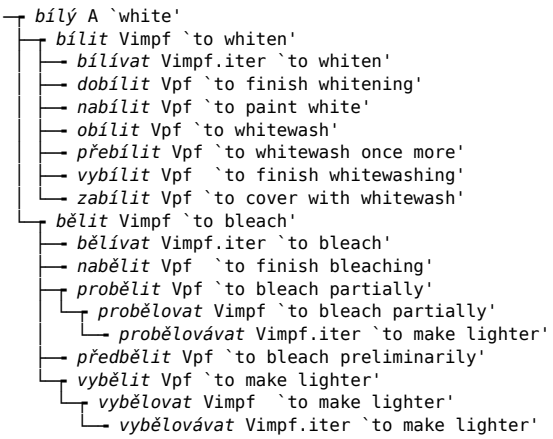


Figure 4. A simplified derivational tree with the root bílý *'white' involving verbs derived directly and indirectly from the adjective, as organized in DeriNet 1.4.*

However, there are still words with alternations that we have not been able to treat so far. The following examples of some significant groups indicate the diversity of problems encountered when extending the coverage of the annotation:

- words with more alternations occurring in a single derivation step; one or more of the alternations usually correlate with alternations in inflection, the other one is in the final grapheme of the stem:

  (82) *vejc-e*$_{\mathrm{N}}$ 'egg' $\xrightarrow{\text{e>a, 0>e, c>č}}$ *vaječ-ný*$_{\mathrm{A}}$ 'made from eggs'
  sníh$_{\mathrm{N}}$ 'snow' $\xrightarrow{\text{í>ě, h>ž}}$ *sněž-ný*$_{\mathrm{A}}$ 'snowy'
  louk-a$_{\mathrm{N}}$ 'meadow' $\xrightarrow{\text{ou>u, k>č}}$ *luč-ní*$_{\mathrm{A}}$ 'meadow'

- words derived from part-of-speech categories that are not contained in DeriNet, particularly from pronouns and numerals:

  (83) *devět*$_{\mathrm{NUM}}$ 'nine' $\xrightarrow{\text{ě>í}}$ *devít-ina*$_{\mathrm{N}}$ '(one) ninth'

- compounds with alternations; a substantial change of the architecture of the database is required in the near future in order to make it possible to represent composition:

  (84) *Bílá*$_{\mathrm{A}}$ *Hora*$_{\mathrm{N}}$ 'White Mountain' (geographical name) $\xrightarrow{\text{í>ě}}$ *běl-o-hor-ský*$_{\mathrm{A}}$ 'from Bílá Hora'

- deverbal nouns are often both formally and semantically based on the whole aspectual pair of verbs (*vysloužit*$_{\mathrm{Vpf}}$ 'to earn', *vysluhovat*$_{\mathrm{Vimpf}}$ 'to earn'); a lin-

guistically adequate solution is to be developed that would enable to connect a word with more than one parent though it is not a compound (without being fused with compounds), etc.

(85) *vy-slouž-i-t*$_{\mathrm{Vpf}}$ 'to earn' $\xrightarrow{\text{y>ý, ou>u}}$ *vý-služ-ba*$_{\mathrm{N}}$ 'retirement' and/or
vy-sluh-ova-t$_{\mathrm{Vimpf}}$ 'to earn' $\xrightarrow{\text{y>ý, h>ž}}$ *vý-služ-ba*$_{\mathrm{N}}$ 'retirement'

The approach to alternations in DeriNet is to be interpreted as the first step in the data-based description of alternations in Czech derivation. The next step is the automatic morphemic segmentation, which makes it possible to look at alternations in connection with particular morphemes. The DeriNet data are expected to be helpful in developing the tools for morphemic segmentation, which is still missing for Czech. For instance, consonant alternations can be detected as an important formal feature indicating the root-suffix (ex. (86)) or suffix-suffix boundary (ex. (87)) while vowel lengthening typically at the second position in nouns derived from verbs (ex. (88)) delimits the prefix-root boundary.

(86) *such-ý*$_{\mathrm{A}}$ 'dry' $\xrightarrow{\text{ch>š}}$ *suš-i-t*$_{\mathrm{V}}$ 'to dry'

(87) *čern-och*$_{\mathrm{N}}$ 'black person' $\xrightarrow{\text{ch>š}}$ *čern-ouš-ek*$_{\mathrm{N}}$ 'black person (demin.)'

(88) *vy-robit*$_{\mathrm{V}}$ 'to product' $\xrightarrow{\text{y>ý}}$ *vý-roba*$_{\mathrm{N}}$ 'production'

Acknowledgements

The research reported on in the present paper was supported by the grant No. GA16-18177S of the Czech Science Foundation.

Bibliography

Aronoff, M. *Word Formation in Generative Grammar*. The MIT Press, Cambridge, 1976.

Aronoff, M. *Morphology by itself: stems and inflectional classes*. The MIT Press, Cambridge, 1994.

Baayen, R., R. Piepenbrock, and L. Gulikers. CELEX2 lexical database. LDC96L14. Philadelphia: Linguistic Data Consortium, 1995.

Baerman, M. *The Oxford Handbook of Inflection*. Oxford University Press, Oxford, 2015.

Baranes, M. and B. Sagot. A Language-independent Approach to Extracting Derivational Relations from an Inflectional Lexicon. In *Proceedings of the 9th International Conference on Language Resources and Evaluation*, pages 2793–2799, Reykjavik, Iceland, May 2014. ISBN 978-2-9517408-8-4.

Bauer, L. *English Word-formation*. Cambridge University Press, Cambridge, 1983.

Bauer, L. Derivational paradigms. In Booij, G. and J. van Marle, editors, *Yearbook of Morphology 1996*, pages 234–256. Kluwer, Dordrecht, 1997.

Bermúdez-Otero, R. and A. McMahon. English Phonology and Morphology. In Aarts, B. and A. McMahon, editors, *The Handbook of English Linguistics*, pages 382–410. Wiley-Blackwell, 2006.

Bojar, O., Z. Žabokrtský, O. Dušek, P. Galuščáková, M. Majliš, D. Mareček, J. Maršík, M. Novák, M. Popel, and A. Tamchyna. The Joy of Parallelism with CzEng 1.0. In *Proceedings of LREC 2012*, pages 3921–3928, Istanbul, Turkey, 2012. ELRA.

Bonami, O., G. Boyé, and F. Kerleroux. L'allomorphie radicale et la relation flexion-construction. In Fradin, Bernard, Françoise Kerleroux, and Marc Plénat, editors, *Aperçus de morphologie du français*, pages 103–125. Presses universitaires de Vincennes, Saint-Denis, 2009.

Booij, G. E. Paradigmatic morphology. In Fradin, Bernard, editor, *La raison morphologique. Hommage á la mémoire de Danielle Corbin*, pages 29–38. Benjamins, Amsterdam, 2008.

Buzássyová, K. *Sémantika slovenských deverbatív*. Veda, Bratislava, 1974.

Bybee, J. L. *Phonology and Language Use*. Cambridge University Press, Cambridge, 2001.

Bybee, J. L. *Frequency of use and the organization of language*. Oxford University Press, New York, 2007.

Bybee, J. L. and M. A. Brewer. Explanation in morphophonemics: Changes in Provençal and Spanish preterite forms. *Lingua*, 52:201–242, 1980.

Comrie, B. *Aspect*. Cambridge University Press, Cambridge, 1976.

Cvrček, V. and P. Vondřička. Nástroj pro slovotvornou analýzu jazykového korpusu. In *Gramatika a korpus*, Hradec Králové, 2013. Gaudeamus.

Čermák, F. *Morfématika a slovotvorba češtiny*. NLN, Prague, 2012.

Daneš, F., M. Dokulil, and J. Kuchař. *Tvoření slov v češtině 2: Odvozování podstatných jmen*. Academia, Prague, 1967.

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

Dokulil, M. The Prague School's Theoretical and Methodological Contribution to "Word Formation" (Derivology). In Luelsdorff, P. A., editor, *The Prague School of Structural and Functional Linguistics*, pages 123–162. Benjamins, Amsterdam, 1994.

Dokulil, M., K. Horálek, J. Hůrková, M. Knappová, and J. Petr. *Mluvnice češtiny 1*. Academia, Prague, 1986.

Furdík, J. *Slovenská slovotvorba*. Náuka, Prešov, 2004.

Gebauer, Jan. *Historická mluvnice jazyka českého*. Academia, Praha, 1984–1929.

Grzegorczykowa, R., R. Laskowski, and H. Wróble. *Gramatyka współczesnego jezyka polskiego. Morfologia*. WN PWN, Warszawa, 1998.

Hajič, J. *Disambiguation of Rich Inflection: Computational Morphology of Czech*. Karolinum Press, Prague, 2004. ISBN 9788024602820.

Hajič, J. and J. Hlaváčová. MorfFlex CZ. LINDAT/CLARIN digital library at Institute of Formal and Applied Linguistics, Charles University in Prague, 2013. `http://hdl.handle.net/11858/00-097C-0000-0015-A780-9`.

Hajič, J., J. Panevová, E. Hajičová, P. Sgall, P. Pajas, J. Štěpánek, J.í Havelka, M. Mikulová, Z. Žabokrtský, and M. Ševčíková-Razímová. Prague Dependency Treebank 2.0. LDC2006T01. Linguistic Data Consortium, Philadelphia, USA, 2006.

Hana, J., D. Zeman, J. Hajič, H. Hanová, B. Hladká, and E. Jeřábek. *Manual for Morphological Annotation, Revision for the Prague Dependency Treebank 2.0*. UFAL MFF UK, Prague, 2005.

Hansen, K. Wortbildung. In Hansen, B., K. Hansen, A. Neubert, and M. Schentke, editors, *Englische Lexikologie*, pages 27–152. VEB Verlag, Leipzig, 1985.

Haspelmath, M. and A. D. Sims. *Understanding Morphology*. Hodder Education, London, 2010.

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

Horecký, J., K. Buzássyová, and J. Bosák. *Dynamika slovnej zásoby súčasnej slovenčiny*. Veda, Bratislava, 1989.

Karlík, P., M. Nekula, and Z. Rusínová. *Příruční mluvnice češtiny*. NLN, Prague, 2000.

Komárek, M. *Příspěvky k české morfologii*. Periplum, Olomouc, 2006.

Kopečný, F. *Slovesný vid v češtině*. Nakladatelství ČSAV, Praha, 1962.

Křen, M., V. Cvrček, T. Čapka, A. Čermáková, M. Hnátková, L. Chlumská, T. Jelínek, D. Kováříková, V. Petkevič, P. Procházka, H. Skoumalová, M. Škrabal, P. Truneček, P. Vondřička, and A. Zasina. SYN2015: A Representative Corpus of Written Czech. UCNK FF UK, Prague, 2015. URL `http://www.korpus.cz`.

Lamprecht, A., D. Šlosar, and J. Bauer. *Historická mluvnice češtiny*. SPN, Praha, 1986.

Lieber, R. and P. Štekauer. *The Oxford Handbook of Derivational Morphology*. Oxford University Press, Oxford, 2014.

Litta, E., M. Passarotti, and C. Culy. Formatio formosa est. Building a Word Formation Based Lexicon for Latin. In *Proceedings of the Third Italian Conference on Computational Linguistics (CLiC-it 2016)*, pages 185–189, Naples, 2016.

Lopatková, M., V. Kettnerová, E. Bejček, A. Vernerová, and Z. Žabokrtský. VALLEX 3.0 – Valency Lexicon of Czech Verbs. Institute of Formal and Applied Linguistics, Charles University in Prague, 2015.

Matthews, P. H. *The Concise Oxford Dictionary of Linguistics*. Oxford University Press, Oxford, 2007.

Mel'čuk, I. A. *Cours de morphologie générale. Vol. 2*. Presses de l'Université de Montréal, Montréal, 1976.

Osolsobě, K. *Algoritmický popis české formální morfologie a strojový slovník češtiny*. PhD thesis, Masarykova univerzita, Brno, 1996.

Osolsobě, K. Alternace hlásková. In Karlík, P., M. Nekula, and J. Pleskalová, editors, *Encyklopedický slovník češtiny*, pages 35–36. NLN, Praha, 2002.

Osolsobě, K. *Česká morfologie a korpusy*. Karolinum, Praha, 2014.

Osolsobě, K. Korpusy jako zdroje dat pro úpravy nástrojů automatické morfologické analýzy. *Časopis pro moderní filologii*, 97:136–145, 2015.

Osolsobě, K., D. Hlaváčková, K. Pala, and P. Šmerk. Exploring Derivational Relations in Czech with the Deriv Tool. In *NLP, Corpus Linguistics, Corpus Based Grammar Research*, pages 152–161, Bratislava, Slovakia, 2009. Tribun. ISBN 978-80-7399-875-2.

Pala, K. and D. Hlaváčková. Derivational Relations in Czech WordNet. In *Proceedings of the 2007 ACL Workshop on Balto-Slavonic Natural Language Processing*, pages 75–81, Prague, 2007.

Pala, K. and P. Smrž. Building Czech Wordnet. *Romanian Journal of Information Science and Technology*, 7:79–88, 2004.

Pala, K. and P. Šmerk. Derivancze – Derivational Analyzer of Czech. In *Text, Speech, and Dialogue*, volume 9302 of *Lecture Notes in Computer Science*, pages 515–523. Springer International Publishing, 2015. doi: 10.1007/978-3-319-24033-6_58.

Pognan, P. and J. Panevová. Génération automatique de lexèmes slaves à partir de leurs racines historiques. Une des bases de l'enseignement multilingue des langues slaves de l'Ouest (Nord et Sud). *Linguistica*, 52:59–75, 2013.

Pounder, A. *Processes and Paradigms in Word-Formation Morphology*. Mouton de Gruyter, Berlin, 2000.

Razímová, M. and Z. Žabokrtský. Annotation of Grammatemes in the Prague Dependency Treebank 2.0. In *Proceedings of the LREC 2006 Workshop on Annotation Science*, pages 12–19, 2006.

Rubenstein, H. *A Comparative Study of Morphophonemic Alternations in Standard Serbo-Croatian, Czech and Russian*. Columbia University, 1950.

Scheer, T. and M. Ziková. The Havlík Pattern and Directional Lower. In *Formal Approaches to Slavic Linguistics. The Second Cornell Meeting*, pages 471–486, Michigan Slavic Publications, 2010.

Scheer, T., A. Starčević, and M. Ziková. Not all zeros are the same and not all alternating vowels are the same. In *Formal Approaches to Slavic Linguistics (FASL 20)*, pages 305–319, Cambridge, Mass., 2011.

Sedláček, R. *Morphematic analyser for Czech*. PhD thesis, Masarykova univerzita, Brno, 2004.

Sedláček, R. and P. Smrž. A New Czech Morphological Analyzer ajka. In *LNCS / Lecture Notes in Artificial Intelligence. Proceedings of the 4th International Conference Text, Speech and Dialogue (TSD 2001)*, pages 100–107, Berlin, 2001. Springer.

Slavíčková, E. *Retrográdní morfematický slovník češtiny*. Academia, Praha, 1975.

Stankiewicz, E. The Consonantal Alternations in the Slavic Declensions. *Word*, 16:183–203, 1960.

Stankiewicz, E. *The Slavic Languages. Unity in Diversity*. Mouton de Gruyter, Berlin, 1986.

Šefčík, O. Ablaut. In Karlík, P., M. Nekula, and J. Pleskalová, editors, *Nový encyklopedický slovník češtiny*, pages 22–24. NLN, Praha, 2016a.

Šefčík, O. Alternace. In Karlík, P., M. Nekula, and J. Pleskalová, editors, *Nový encyklopedický slovník češtiny*, pages 87–88. NLN, Praha, 2016b.

Ševčíková, M. and Z. Žabokrtský. AdjDeriNet. LINDAT/CLARIN digital library at Institute of Formal and Applied Linguistics, Charles University in Prague, 2014a. URL `http://hdl.handle.net/11234/1-1467`.

Ševčíková, M. and Z. Žabokrtský. Word-Formation Network for Czech. In *Proceedings of the 9th International Conference on Language Resources and Evaluation*, pages 1087–1093, Reykjavik, Iceland, May 2014b. ISBN 978-2-9517408-8-4.

Ševčíková, M., J. Panevová, and P. Pognan. Inflectional and derivational paradigm of verbs in Czech: the role of the category of aspect. In *Abstracts from the First Workshop on Paradigmatic Word Formation Modeling*. Université Toulouse, 2017, in press.

Šimandl, J. *Slovník afixů užívaných v češtině*. Karolinum, Praha, 2016.

Šiška, Z. *Bázový morfematický slovník češtiny*. UPOL, Olomouc, 2005.

Šmerk, P. Fast Morphological Analysis of Czech. In *Proceedings of Third Workshop on Recent Advances in Slavonic Natural Language Processing (RASLAN 2007)*, pages 13–16, Brno, 2007. Masaryk University.

Šnajder, J. 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, Reykjavik, Iceland, May 2014. ISBN 978-2-9517408-8-4.

Štekauer, P. *An Onomasiological Theory of English Word-Formation*. Benjamins, Amsterdam, 1998.

Štekauer, P., S. Valera, and L. Körtvélyessy. *Word-Formation in the World's Languages*. Cambridge University Press, Cambridge, 2012.

Štícha, F. *Akademická gramatika spisovné češtiny*. Academia, Praha, 2013.

Švedova, N. J. *Russkaja gramatika. Vol. 1*. Nauka, Moskva, 1980.

Večerka, R. Palatalizace. In Karlík, P., M. Nekula, and J. Pleskalová, editors, *Nový encyklopedický slovník češtiny*, pages 1287–1288. NLN, Praha, 2016.

Vey, M. Les préverbes vides en tchèque moderne. *Revue des Études de Slave*, 29:82–107, 1952.

Vidra, J., Z. Žabokrtský, M. Ševčíková, and M. Straka. DeriNet 1.0. LINDAT/CLARIN digital library at Institute of Formal and Applied Linguistics, Charles University in Prague, 2015. URL `http://hdl.handle.net/11234/1-1520`.

Vidra, J., Z. Žabokrtský, M. Ševčíková, and M. Straka. DeriNet 1.2. LINDAT/CLARIN digital library at Institute of Formal and Applied Linguistics, Charles University in Prague, 2016. URL `http://hdl.handle.net/11234/1-1807`.

Zeller, B., J. Šnajder, and S. Padó. DErivBase: Inducing and Evaluating a Derivational Morphology Resource for German. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, pages 1201–1211, Sofia, Bulgaria, 2013.

Ziková, M. Morfonologické alternace v současné češtině. In Ulilčný, O., editor, *Preliminária k moderní mluvnici češtiny*, pages 177–201. UPOL, Olomouc, 2015.

Ziková, M. Alternace kvantity. In Karlík, P., M. Nekula, and J. Pleskalová, editors, *Nový encyklopedický slovník češtiny*, pages 88–90. NLN, Praha, 2016a.

Ziková, M. Alternace vokálů s nulou. In Karlík, P., M. Nekula, and J. Pleskalová, editors, *Nový encyklopedický slovník češtiny*, pages 90–92. NLN, Praha, 2016b.

Žabokrtský, Z., M. Ševčíková, M. Straka, J. Vidra, and A. Limburská. Merging Data Resources for Inflectional and Derivational Morphology in Czech. In *Proceedings of the 10th International Conference on Language Resources and Evaluation*, pages 1307–1314, Portoroz, Slovenia, 2016.

Appendix: Morphographemic alternations in contemporary Czech

	altern.		example	counter-example
(A, B, C) vowel alternations	a>á (bidir.)	in pref.	*za-bal-i-t → zá-bal* 'to pack' 'wet pack'	*na-hr-á-t → na-hr-á-vka* 'to record' 'recording'
		in root	*vrat-a → vrát-ka* 'gate' (dimin.)	*pat-a → pat-ka* 'heel' (dimin.)
		in suf.	*hled-a-t → hled-á-ní* 'to search' 'search'	*hled-a-t → hled-a-ný* 'to search' 'searched'
	a>e (bidir.)	in root	*úřad → úřed-ník* 'office' 'officer'	*ná-klad → ná-klad-ní* 'load' 'cargo'
	a>ě	in root	*svat-ý → svět-ec* 'holy' 'holy man'	
	a>o	in root	*hrab-a-t → hrob* 'to dig' 'grave'	*s-pad-a-t → s-pad* 'to fall down' 'fallout'
	á>a (bidir.)	in root	*kámen → kamen-ný* 'stone' 'stony'	*památ-ka → památ-ný* 'memory' 'memorable'
		in suf.	*ps-á-t → ps-a-ní* 'to write' 'letter'	
	á>e (bidir.)	in root	*o-třás-t → o-třes* 'to shake' 'shake'	*krás-t → krád-ež* 'to steal' 'theft'
	á>i	in root	*ďábel → dibl-ík* 'devil' 'imp'	*ďábel → ďáblík* 'devil' (dimin.)
	á>í (bidir.)	in root	*přá-t → pří-tel* 'to wish' 'friend'	*hrá-t → hrá-č* 'to play' 'player'
	e>a (bidir.)	in root	*vejc-e → vaječ-ný* 'egg' 'made from eggs'	*strejc → strejc-ův* 'uncle' 'uncle's'
	e>á (bidir.)	in root	*deset → desát-ý* 'ten' 'tenth'	
	e>é (bidir.)	in root	*oheň → ohén-ek* 'fire' (dimin.)	*ú-čes → ú-čes-ek* 'hairstyle' (dimin.)
		in suf.	*prst-en → prst-én-ek* 'ring' (dimin.)	
	e>o	in root	*lež-e-t → po-lož-i-t* 'to lie' 'to lay down'	*žel-e-t → o-žel-e-t* 'to regret' 'to do without'
	e>í (bidir.)	in root	*deset → desít-ka* 'ten' '(number) ten'	*raket-a → raket-ka* 'rocket' (dimin.)
	e>ý	in root	*postel → postýl-ka* 'bed' (dimin.)	*činel → činel-ek* 'cymbal' (dimin.)
		in suf.	*uči-tel → uči-týl-ek* 'teacher' (dimin.)	
	ě>í	in root	*květ → kvít-ek*	*paměť → pamět-ník*

altern.		example	counter-example
(bidir.)		'blossom' (dimin.)	'memory' 'survivor'
ě>á (bidir.)	in root	*paměť → památ-ka* 'memory' 'souvenir'	*Bět-a → Bět-ka* fem. name (dimin.)
é>a	in root	*vléc-t → vlak* 'to pull' 'train'	*vléc-t → vlek* 'to pull' 'ski lift'
é>e (bidir.)	in root	*lét-o → let-ní* 'summer' 'summer-adj'	*bazén → bazén-ek* 'pool' (dimin.)
é>í	in root	*mléko → mlíko* 'milk' '(non-stand.)'	*mléko → mléčný* 'milk' 'milk'
	in suf.	*svíc-en → svíc-ín-ek* 'candlestick' (dimin.)	
é>ý	in root	*okén-ko → okýn-ko* 'window' '(non-stand.)'	
	in suf.	*prst-en → prst-ýn-ek* 'ring' (dimin.)	*prst-en → prst-en-ec* 'ring' 'big ring'
i>e	insuf.	*zlob-i-t → zlob-e-ní* 'to misbehave' 'misbehavior'	*zlob-i-t → zlob-i-vý* 'to misbehave' 'naughty'
i>í (bidir.)	in pref.	*při-děl-i-t → pří-děl* 'to assign' 'ration'	*při-hr-á-t → při-hr-á-vka* 'to pass' 'pass'
	in root	*list → líst-ek* 'leaf' (dimin.)	*sešit → sešit-ek* 'block' (dimin.)
	in suf.	*čum-il → čum-íl-ek* 'gaper' (dimin.)	*text-il → text-il-ka* 'textile' 'textile factory'
í>i (bidir.)	in root	*líp-a → lip-ka* 'linden' (dimin.)	*píst → píst-ek* 'piston' (dimin.)
í>a	in root	*žít → žat-va* 'to mow' 'mowing'	
í>á (bidir.)	in root	*přítel → přátel-ský* 'friend' 'friendly'	
í>e (bidir.)	in root	*říd-i-t → řed-i-tel* 'to lead' 'director'	*z-říd-i-t → z-říz-en-ec* 'to establish' 'attendant'
	in suf.	*zaj-íc → zaj-eč-í* 'hare' 'hare's'	*měs-íc → měs-íč-ní* 'moon' 'lunar'
í>ě (bidir.)	in root	*vítr → větr-ný* 'wind' 'windy'	*mír-a → mír-ný* 'degree' 'moderate'
o>á	in root	*s-klon-i-t → s-klán-ě-t* 'to incline$_{\mathrm{pf}}$' 'to incline$_{\mathrm{impf}}$'	
o>ó (bidir.)	in root	*Božen-a → Bóž-a* (fem. name) (familiar)	*Božen-a → Bož-ka* (fem. name) (familiar)
o>ů (bidir.)	in pref.	*pro-střel-i-t → prů-střel* 'to shoot through' 'shot through'	*pro-slov-i-t → pro-slov* 'to give a speech' 'speech'
	in root	*cop → cůp-ek* 'plait' (dimin.)	*strom → stromek* 'tree' (dimin.)

	altern.		example	counter-example
		in suf.	*lib-ost* → *lib-ůst-ka* 'liking' (dimin.)	
	o>ou	in root	*boř-i-t* → *bour-a-t* 'to destroy$_{\text{pf}}$' 'to destroy$_{\text{impf}}$'	*po-noř-i-t* → *po-noř-ova-t* 'to dip$_{\text{pf}}$' 'to dip$_{\text{impf}}$'
		in suf.	*čern-och* → *čern-ouš-ek* 'black man' (dimin.)	*let-os* → *let-oš-ek* 'this year' 'this year'
	ó>o (bidir.)	in root	*próz-a* → *proz-aický* 'prose' 'prosaic'	
	ou>u (bidir.)	in root	*kouř-i-t* → *kuř-ák* 'to smoke' 'smoker'	*bouř-i-t* → *bouř-e* 'to storm' 'storm'
		in suf.	*ln-ou-t* → *ln-u-tí* 'to adhere' 'adhering'	
	u>ou (bidir.)	in root	*dub* → *doub-ek* 'oak' (dimin.)	*stuh-a* → *stuž-ka* 'ribbon' (dimin.)
	u>ú (bidir.)	in pref.	*u-lovit* → *ú-lovek* 'to catch' 'catch'	*ú-toč-i-t* → *ú-tok* 'to attack' 'attack'
	ú>u (bidir.)	in pref.	*ú-cta* → *u-ctivý* 'respect' 'respectful'	*ú-nav-a* → *ú-nav-ný* 'fatique' 'tiring'
		in root	*úz-ký* → *uz-oučký* narrow' (dimin.)	*útl-ý* → *útl-oučký* 'thin' (dimin.)
	ů>o (bidir.)	in root	*kůž-e* → *kož-ený* 'leather' 'leather-adj'	*kůr-a* → *kůr-ový* 'bark' 'bark-adj.'
	y>ý (bidir.)	in pref.	*vy-br-a-t* → *vý-bor* 'to choose' 'board'	*vy-hlás-i-t* → *vy-hláš-ka* 'to declare' 'notice'
		in root	*vys-oký* → *výš-ka* 'high' 'height'	*ryb-a* → *ryb-ka* 'fish' (dimin.)
	ý>y (bidir.)	in root	*hýb-a-t* → *hyb-ný* 'to move' 'movable'	*hýb-a-t* → *hýb-ací* 'to move' 'moving'
(D) vowel deletion	e>0 (bidir.)	in root	*kart-a* → *karet-ní* 'card' 'card-adj'	*nárt* → *nárt-ní* 'instep' 'instep-adj.'
		in suf.	*dár-ek* → *dár-k-ový* 'gift' 'gift-adj'	*do-tek* → *do-tek-ový* 'touch' 'touch-adj.'
	é>0 (bidir.)	in root	*déšť* → *dšt-í-t* 'rain' 'to rain'	
	u>0	in root	*such-ý* → *sch-nout* 'dry' 'to dry'	
(E) vowel insertion	0>e (bidir.)	in root	*hr-a* → *her-ní* 'play' 'playing'	*hr-á-t* → *hr-a* 'to play' 'play'
		in suf.	*služ-b-a* → *služ-eb-ní* 'service' 'business-adj'	
	0>é (bidir.)	in root	*okn-o* → *okén-ko* 'window' (dimin.)	
	0 >o	in root	*hřm-ít* → *hrom*	

	altern.		example	counter-example
			‘to thunder’ ‘thunder’	
	0>i	in root	*na-ps-a-t* → *ná-pis* ‘to write’ ‘sign’	
	0>í	in root	*ps-á-t* → *pís-ař* ‘to write’ ‘writer’	
	0>y	in root	*za-mk-nou-t* → *za-myk-a-t* ‘to lock$_{\mathrm{pf}}$’ ‘to lock$_{\mathrm{impf}}$’	
	0>ý	in root	*na-zv-a-t* → *na-zýv-a-t* ‘to call$_{\mathrm{pf}}$’ ‘to call$_{\mathrm{impf}}$’	
(F) consonant alternations	c>č	in root	*ovc-e* → *ovč-í* ‘sheep’ ‘sheep’s’	
		in suf.	*chlap-ec* → *chlap-eč-ek* ‘boy’ (dimin.)	
	c>k (bidir.)	in root	*péc-t* → *pek-ař* ‘to bake’ ‘baker’	*pec* → *pec-ař* ‘oven’ ‘oven builder’
	č>k (bidir.)	in root	*breč-e-t* → *brek* ‘to cry’ ‘cry’	
	d>ď (bidir.)	in root	*hněd-ý* → *hněď* ‘brown’ ‘brown (colour)’	*sled-ova-t* → *sled* ‘to follow’ ‘sequence’
	d>z	in root	*tvrd-ý* → *tvrz* ‘hard’ ‘fort’	*hod-i-t* → *hod* ‘to throw’ ‘throw’
	ď>d (bidir.)	in root	*loď* → *lod-ní* ‘ship’ ‘shipping’	
	g>ž	in root	*chirurg* → *chirurž-ka* ‘surgeon’ ‘woman surgeon’	
	h>z	in root	*tuh-ý* → *tuz-e* ‘solid’ ‘solid(ly)’	
	h>ž (bidir.)	in root	*sníh* → *sníž-ek* ‘snow’ (dimin.)	
	ch>š (bidir.)	in root	*živočich* → *živočiš-ný* ‘animal’ ‘animal-adj’	*všechen* → *po-všech-ný* ‘all’ ‘general’
		in suf.	*čern-och* → *čern-oš-ka* ‘black man’ ‘black woman’	
	k>c (bidir.)	in root	*trpk-ý* → *trpc-e* ‘bitter’ ‘bitterly’	
		in suf.	*blíz-k-ý* → *blíz-c-e* ‘close’ ‘closely’	
	k>č (bidir.)	in root	*ruk-a* → *ruč-ní* ‘hand’ ‘manual’	
		in suf.	*balet-k-a* → *balet-č-in* ‘ballerina’ ‘ballerina’s’	
	k>t	in root	*hrušk-a* → *hrušt-ička*	

	altern.		example	counter-example
			‘pear’ (dimin.)	
		in suf.	*služ-k-a → služ-t-ička*	
			‘housemaid’ (dimin.)	
	n>ň	in root	*čern-ý → čerň*	*u-hrn-ou-t → ú-hrn*
	(bidir.)		‘black’ ‘black (colour)’	‘to sum up’ ‘summary’
		in suf.	*železn-ý → železň-ák*	
			‘iron’ ‘basalt’	
	ň>n	in root	*skříň → skřín-ka*	
	(bidir.)		‘closet’ (dimin.)	
	r>ř	in root	*star-ý → stař-ík*	
	(bidir.)		‘old’ ‘old man’	
	ř>r	in suf.	*truhl-ář → truhl-ár-na*	
	(bidir.)		‘joiner’ ‘joiner’s shop’	
	s>š	in root	*mysl-e-t → myšl-e-ní*	
			‘to think’ ‘thinking’	
	š>ch	in root	*prš-e-t → s-prch-a*	*srš-e-t → srš-atý*
	(bidir.)		‘to rain’ ‘shower’	‘to fume’ ‘furious’
	t>ť	in root	*žlut-ý → žluť*	
	(bidir.)		‘yellow’ ‘yellow (colour)’	
	t>c	in root	*svít-i-t → svíc-e*	
			‘to shine’ ‘candle’	
		in suf.	*o-boh-at-i-t → o-boh-ac-ova-t*	*boh-at-ý → boh-at-ec*
			‘to enrich$_{\mathrm{pf}}$’ ‘to enrich$_{\mathrm{impf}}$’	‘rich’ ‘rich man’
	ť>t	in root	*řiť → řit-ní*	
	(bidir.)		‘anus’ ‘anal’	
	z>ž	in root	*řez-a-t → řež*	*řez-a-t → řez*
			‘to cut’ ‘scuffle’	‘to cut’ ‘section’
	ž>h	in root	*slouž-i-t → sluh-a*	*těž-i-t → těž-ba*
	(bidir.)		‘to serve’ ‘servant’	‘to mine’ ‘mining’
(G) cons. del. and ins.	k>0	in root	*Hamburk → hamburs-ký*	
			‘Hamburg’ ‘from Hamburg’	
	g>0	in root	*Peking → pekin-ský*	
			‘Beijing’ ‘from Beijing’	
	p>0	in root	*kyp-ě-t → ky-nou-t*	*máv-a-t → máv-nou-t*
			‘to brim’ ‘to rise’	‘to wave$_{\mathrm{impf}}$’ ‘to wave$_{\mathrm{pf}}$’
	v>0	in root	*kýv-a-t → ky-nou-t*	
			‘to nod’ ‘to wave’	
	0>j	in root	*mí-t → jmě-ní*	
			‘to have’ ‘property’	
(H) group alternations	ck>čť	root/suf.	*řec-k-ý → řeč-t-ina*	
			‘Greek’ ‘Greek lang.’	
	sk>šť	root/suf.	*rus-k-ý → ruš-t-ina*	

	altern.		example	counter-example
		in suf.	'Russian' 'Russian lang.' *arab-sk-ý* → *arab-št-ina* 'Arabic' 'Arabic lang.'	
	st>šť	in root	*měst-o* → *měšť-an* 'town' 'burgher'	*chvost* → *chvost-an* 'tail' 'saki monkey'
(I) mixed alternations	á>av	in root	*stá-t* → *stav* 'to stand' 'to state'	*stá-t* → *stá-va-t* 'to stand$_{\mathrm{impf}}$' 'to stand$_{\mathrm{iter}}$'
	á>ěj	in root	*vá-t* → *věj-íř* 'to blow' 'fan'	*vá-t* → *vá-nice* 'to blow' 'blizzard'
	á>oj	in root	*stá-t* → *stoj-ící* 'to stand' 'standing'	*stá-t* → *stá-va-t* 'to stand$_{\mathrm{impf}}$' 'to stand$_{\mathrm{iter}}$'
	á>av	in root	*stá-t* → *stav-ba* 'to stand' 'building'	*stá-t* → *stá-va-t* 'to stand$_{\mathrm{impf}}$' 'to stand$_{\mathrm{iter}}$'
	í>ij	in root	*bí-t* → *bij-ící* 'to beat' 'beating'	
	í>oj	in root	*bí-t* → *boj* 'to beat' 'fight'	
	ý>ov	in root	*krý-t* → *krov* 'to cover' 'roof frame'	

Address for correspondence:
Magda Ševčíková
sevcikova@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 110 APRIL 2018 43–70

Training Tips for the Transformer Model

Martin Popel, Ondřej Bojar

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

Abstract

This article describes our experiments in neural machine translation using the recent Tensor2Tensor framework and the Transformer sequence-to-sequence model (Vaswani et al., 2017). We examine some of the critical parameters that affect the final translation quality, memory usage, training stability and training time, concluding each experiment with a set of recommendations for fellow researchers. In addition to confirming the general mantra "more data and larger models", we address scaling to multiple GPUs and provide practical tips for improved training regarding batch size, learning rate, warmup steps, maximum sentence length and checkpoint averaging. We hope that our observations will allow others to get better results given their particular hardware and data constraints.

1. Introduction

It has been already clearly established that neural machine translation (NMT) is the new state of the art in machine translation, see e.g. the most recent evaluation campaigns (Bojar et al., 2017a; Cettolo et al., 2017). Many fundamental changes of the underlying neural network architecture are nevertheless still frequent and it is very difficult to predict which of the architectures has the best combination of properties to win in the long term, considering all relevant criteria like translation quality, model size, stability and speed of training, interpretability but also practical availability of good implementations. A considerable part of a model's success in translation quality consists in the training data, the model's sensitivity to noise in the data but also on a wide range of hyper-parameters that affect the training. Having the right setting of them turns out to be often a critical component for the success.

 Corresponding author: popel@ufal.mff.cuni.cz
Cite as: Martin Popel, Ondřej Bojar. Training Tips for the Transformer Model. The Prague Bulletin of Mathematical Linguistics No. 110, 2018, pp. 43–70. doi: 10.2478/pralin-2018-0002.

In this article, we experiment with a relatively new NMT model, called Transformer (Vaswani et al., 2017) as implemented in the Tensor2Tensor[1] (abbreviated T2T) toolkit, version 1.2.9. The model and the toolkit have been released shortly after the evaluation campaign at WMT2017[2] and its behavior on large-data news translation is not yet fully explored. We want to empirically explore some of the important hyper-parameters. Hopefully, our observations will be useful also for other researchers considering this model and framework.

While investigations into the effect of hyper-parameters like learning rate and batch size are available in the deep-learning community (e.g. Bottou et al., 2016; Smith and Le, 2017; Jastrzebski et al., 2017), these are either mostly theoretic or experimentally supported from domains like image recognition rather than machine translation. In this article, we fill the gap by focusing exclusively on MT and on the Transformer model only, providing hopefully the best practices for this particular setting.

Some of our observations confirm the general wisdom (e.g. larger training data are generally better) and quantify the behavior on English-to-Czech translation experiments. Some of our observations are somewhat surprising, e.g. that two GPUs are more than three times faster than a single GPU, or our findings about the interaction between maximum sentence length, learning rate and batch size.

The article is structured as follows. In Section 2, we discuss our evaluation methodology and main criteria: translation quality and speed of training. Section 3 describes our dataset and its preparations. Section 4 is the main contribution of the article: a set of commented experiments, each with a set of recommendations. Finally, Section 5 compares our best Transformer run with systems participating in WMT17. We conclude in Section 6.

2. Evaluation Methodology

Machine translation can be evaluated in many ways and some forms of human judgment should be always used for the ultimate resolution in any final application. The common practice in MT research is to evaluate the model performance on a test set against one or more human reference translations. The most widespread automatic metric is undoubtedly the BLEU score (Papineni et al., 2002), despite its acknowledged problems and better-performing alternatives (Bojar et al., 2017b). For simplicity, we stick to BLEU, too (we evaluated all our results also with chrF (Popović, 2015), but found no substantial differences from BLEU). In particular, we use the case-insensitive sacréBLEU[3] which uses a fixed tokenization (identical to `mteval-v14.pl --interna-`

[1] `https://github.com/tensorflow/tensor2tensor`

[2] `http://www.statmt.org/wmt17`

[3] `https://github.com/awslabs/sockeye/tree/master/contrib/sacrebleu`
The signature of the BLEU scores reported in this paper is `BLEU+case.lc+lang.en-cs+numrefs.1+smooth.exp+test.wmt13+tok.intl+version.1.2.3`.

`tional-tokenization`) and automatically downloads the reference translation for a given WMT testset.

2.1. Considerations on Stopping Criterion

The situation in NMT is further complicated by the fact that the training of NMT systems is usually non-deterministic,[4] and (esp. with the most recent models) hardly ever converges or starts overfitting[5] on reasonably big datasets. This leads to learning curves that never fully flatten let alone start decreasing (see Section 4.2). The common practice of machine learning to evaluate the model on a final test set when it started overfitting (or a bit sooner) is thus not applicable in practice.

Many papers in neural machine translation do not specify any stopping criteria whatsoever. Sometimes, they mention only an approximate number of days the model was trained for, e.g. Bahdanau et al. (2015), sometimes the exact number of training steps is given but no indication on "how much converged" the model was at that point, e.g. Vaswani et al. (2017). Most probably, the training was run until no further improvements were clearly apparent on the development test set, and the model was evaluated at that point. Such an approximate stopping criterion is rather risky: it is conceivable that different setups were stopped at different stages of training and their comparison is not fair.

A somewhat more reliable method is to keep training for a specified number of iterations or a certain number of epochs. This is however not a perfect solution either, if the models are not quite converged at that time and the difference in their performance is not sufficiently large. It is quite possible that e.g. a more complex model would need a few more epochs and eventually arrived at a higher score than its competitor. Also, the duration of one training step (or one epoch) differs between models (see Section 4.1) and from the practical point of view, we are mostly interested in the wall-clock time.

When we tried the standard technique of early stopping, when N subsequent evaluations on the development test set do not give improvements larger than a given delta, we saw a big variance in the training time and final BLEU, even for experiments with the same hyper-parameters and just a different random seed. Moreover to get the best results, we would have had to use a very large N and a very small delta.

[4] Even if we fix the random seed (which was not done properly in T2T v1.2.9), a change of some hyper-parameters may affect the results not because of the change itself, but because it influenced the random initialization.

[5] By overfitting we mean here that the translation quality (test-set BLEU) begins to worsen, while the training loss keeps improving.

2.2. Our Final Choice: Full Learning Curves

Based on the discussion above, we decided to report always the full learning curves and not just single scores. This solution does not fully prevent the risk of premature judgments, but the readers can at least judge for themselves if they would expect any sudden twist in the results or not.

In all cases, we plot the case-insensitive BLEU score against the wall-clock time in hours. This solution obviously depends on the hardware chosen, so we always used the same equipment: one up to eight GeForce GTX 1080 Ti GPUs with NVIDIA driver 375.66. Some variation in the measurements is unfortunately unavoidable because we could not fully isolate the computation from different processes on the same machine and from general network traffic, but based on our experiments with replicated experiments such variation is negligible.

2.3. Terminology

For clarity, we define the following terms and adhere to them for the rest of the paper:

Translation quality is an automatic estimate of how well the translation carried out by a particular fixed model expresses the meaning of the source. We estimate translation quality solely by BLEU score against one reference translation.

Training Steps denote the number of iterations, i.e. the number of times the optimizer update was run. This number also equals the number of (mini)batches that were processed.

Batch Size is the number of training examples used by one GPU in one training step. In sequence-to-sequence models, batch size is usually specified as the number of *sentence pairs*. However, the parameter `batch_size` in T2T translation specifies the approximate number of *tokens* (subwords) in one batch.[6] This allows to use a higher number of short sentences in one batch or a smaller number of long sentences.

Effective Batch Size is the number of training examples consumed in one training step. When training on multiple GPUs, the parameter `batch_size` is interpreted per GPU. That is, with `batch_size=1500` and 8 GPUs, the system actually digests 12k subwords of each language in one step.

Training Epoch corresponds to one complete pass over the training data. Unfortunately, it is not easy to measure the number of training epochs in T2T.[7] T2T

[6] For this purpose, the number of tokens in a sentence is defined as the maximum of source and target subwords. T2T also does reordering and bucketing of the sentences by their length to minimize the use of padding symbols. However, some padding is still needed, thus `batch_size` only approximates the actual number of (non-padding) subwords in a batch.

[7] `https://github.com/tensorflow/tensor2tensor/issues/415`

reports only the number of training steps. In order to convert training steps to epochs, we need to multiply the steps by the effective batch size and divide by the number of subwords in the training data (see Section 3.1). The segmentation of the training data into subwords is usually hidden to the user and the number of subwords must be thus computed by a special script.

Computation Speed is simply the observed number of training steps per hour. Computation speed obviously depends on the hardware (GPU speed, GPU-CPU communication) and software (driver version, CUDA library version, implementation). The main parameters affecting computation speed are the model size, optimizer and other settings that directly modify the formula of the neural network.

Training Throughput is the amount of training data digested by the training. We report training throughput in subwords per hour. Training Throughput equals to the Computation Speed multiplied by the effective batch size.

Convergence Speed or **BLEU Convergence** is the increase in BLEU divided by time. Convergence speed changes heavily during training, starting very high and decreasing as the training progresses. A converged model should have convergence speed of zero.

Time Till Score is the training time needed to achieve a certain level of translation quality, in our case BLEU. We use this as an informal measure because it is not clear how to define the moment of "achieving" a given BLEU score. We define it as time after which the BLEU never falls below the given level.[8]

Examples Till Score is the number of training examples (in subwords) needed to achieve a certain level of BLEU. It equals to the Time Till Score multiplied by Training Throughput.

2.4. Tools for Evaluation within Tensor2Tensor

T2T, being implemented in TensorFlow, provides nice TensorBoard visualizations of the training progress. The original implementation was optimized towards speed of evaluation rather than towards following the standards of the field. T2T thus reports "approx-bleu" by default, which is computed on the internal subwords (never exposed to the user, actually) instead of words (according to BLEU tokenization). As a result, "approx-bleu" is usually about 1.2–1.8 times higher than the real BLEU. Due to its dependence on the training data (for the subword vocabulary), it is not easily reproducible in varying experiments and thus not suitable for reporting in publications.

[8] Such definition of Time Till Score leads to a high variance of its values because of the relatively high BLEU variance between subsequent checkpoints (visible as a "flickering" of the learning curves in the figures). To decrease the variation one can use a bigger development test set.

	sentences	EN words	CS words
CzEng 1.7	57 M	618 M	543 M
europarl-v7	647 k	15 M	13 M
news-commentary-v11	190 k	4.1 M	3.7 M
commoncrawl	161 k	3.3 M	2.9 M
Total	58 M	640 M	563 M

Table 1: Training data resources

We implemented a helper script `t2t-bleu` which computes the "real" BLEU (giving the same result as sacréBLEU with `--tokenization intl`). Our script can be used in two ways:

- To evaluate one translated file:
  `t2t-bleu --translation=my-wmt13.de --reference=wmt13_deen.de`
- To evaluate all translations in a given directory (created e.g. by `t2t-translate-all`) and store the results in a TensorBoard events file. All the figures in this article were created this way.

We also implemented `t2t-translate-all` and `t2t-avg-all` scripts, which translate all checkpoints in a given directory and average a window of N subsequent checkpoints, respectively.[9] For details on averaging see Section 4.10.

3. Data Selection and Preprocessing

We focused on the English-to-Czech translation direction. Most of our training data comes from the CzEng parallel treebank, version 1.7 (57M sentence pairs),[10] and the rest (1M sentence pairs) comes from three smaller sources (Europarl, News Commentary, Common Crawl) as detailed in Table 1.

We use this dataset of 58M sentence pairs for most our experiments. In some experiments (in Sections 4.2 and 4.6), we substitute CzEng 1.7 with an older and considerably smaller CzEng 1.0 (Bojar et al., 2012) containing 15M sentence pairs (233M/206M of en/cs words).

To plot the performance throughout the training, we use WMT newstest2013 as a development set (not overlapping with the training data). In Section 5, we apply our best model (judged from the performance on the development set) to the WMT newstest2017, for comparison with the state-of-the-art systems.

[9] All three scripts are now merged in the T2T master. All three scripts can be used while the training is still in progress, i.e. they wait a given number of minutes for new checkpoints to appear.

[10] `http://ufal.mff.cuni.cz/czeng/czeng17`, which is a subset of CzEng 1.6 (Bojar et al., 2016).

3.1. Training Data Preprocessing

Data preprocessing such as tokenization and truecasing has always been a very important part of the setup of statistical machine translation systems. A huge leap in scaling NMT to realistic data size has been achieved by the introduction of subword units (Sennrich et al., 2016), but the long-term vision of the deep-learning community is to leave all these "technicalities" up to the trained neural network and feed it with as original input as possible (see e.g. Lee et al., 2016).

T2T adopts this vision and while it supports the use of external subword units, it comes with its own built-in method similar to the word-piece algorithm by Wu et al. (2016) and does not expect the input to be even tokenized. Based on a small sample of the training data, T2T will train a subword vocabulary and apply it to all the training and later evaluation data.

We follow the T2T default and provide raw plain text training sentences. We use the default parameters: shared source and target (English and Czech) subword vocabulary of size 32k.[11] After this preprocessing, the total number of subwords in our main training data is 992 millions (taking the maximum of English and Czech lengths for each sentence pair, as needed for computing the number of epochs, see Section 2.3). The smaller dataset CzEng 1.0 has 327 million subwords. In both cases the average number of subwords per (space-delimited) word is about 1.5.

Even when following the defaults, there are some important details that should be considered. We thus provide our first set of technical tips here:

Tips on Training Data Preprocessing

- Make sure that the subword vocabulary is trained on a sufficiently large sample of the training data.[12]
- As discussed in Section 4.5, a higher batch size may be beneficial for the training and the batch size can be higher when excluding training sentences longer than a given threshold. This can be controlled with parameter `max_length` (see Section 4.4), but it may be a good idea to exclude too long sentences even before preparing the training data using `t2t-datagen`. This way the TFRecords training files will be smaller and their processing a bit faster.[13]

[11] More details on T2T with BPE subword units by Sennrich et al. (2016) vs. the internal implementation can be found in the technical report "Morphological and Language-Agnostic Word Segmentation for NMT" attached to the Deliverable 2.3 of the project QT21: `http://www.qt21.eu/resources/`.

[12] This is controlled by a `file_byte_budget` constant, which must be changed directly in the source code in T2T v1.2.9. A sign of too small training data for the subword vocabulary is that the `min_count` as reported in the logs is too low, so the vocabulary is estimated from words seen only once or twice.

[13] We did no such pre-filtering in our experiments.

4. Experiments

In this section, we present several experiments, always summarizing the observations and giving some generally applicable tips that we learned. All experiments were done with T2T v1.2.9 unless stated otherwise.

We experiment with two sets of hyper-parameters pre-defined in T2T: `transformer_big_single_gpu` (BIG) and `transformer_base_single_gpu` (BASE), which differ mainly in the size of the model. Note that `transformer_big_single_gpu` and `transformer_base_single_gpu` are just names of a set of hyper-parameters, which can be applied even when training on multiple GPUs, as we do in our experiments, see Section 4.7.[14]

Our baseline setting uses the BIG model with its default hyper-parameters except for:

- `batch_size=1500` (see the discussion of different sizes in Section 4.5),
- `--train_steps=6000000`, i.e. high enough, so we can stop each experiment manually as needed,
- `--save_checkpoints_secs=3600` which forces checkpoint saving each hour (see Section 4.10),
- `--schedule=train` which disables the internal evaluation with `approx_bleu` and thus makes training a bit faster (see Section 2).[15]

4.1. Computation Speed and Training Throughput

We are primarily interested in the translation quality (BLEU learning curves and Time Till Score) and we discuss it in the following sections 4.2–4.10. In this section, we focus however only on the *computation speed* and *training throughput*. Both are affected by three important factors: batch size, number of used GPUs and model size. The speed is usually almost constant for a given experiment.[16]

Table 2 shows the computation speed and training throughput for a single GPU and various batch sizes and model sizes (BASE and BIG). The BASE model allows for using a higher batch size than the BIG model. The cells where the BIG model resulted in out-of-memory errors are marked with "OOM".[17] We can see that the computa-

[14] According to our experiments (not reported here), `transformer_big_single_gpu` is better than `transformer_big` even when training on 8 GPUs, although the naming suggests that the T2T authors had an opposite experience.

[15] Also there are some problems with the alternative schedules `train_and_evaluate` (it needs more memory) and `continuous_train_and_eval` (see `https://github.com/tensorflow/tensor2tensor/issues/556`).

[16] TensorBoard shows `global_step/sec` statistics, i.e. the computation speed curve. These curves in our experiments are almost constant for the whole training with variation within 2%, except for moments when a checkpoint is being saved (and the computation speed is thus much slower).

[17] For these experiments, we used `max_length=50` in order to be able to test bigger batch sizes. However, in additional experiments we checked that `max_length` does not affect the training throughput itself.

	model	
batch_size	BASE	BIG
500	43.4k	23.6k
1000	30.2k	13.5k
1500	22.3k	9.8k
2000	16.8k	7.5k
2500	14.4k	6.5k
3000	12.3k	OOM
4500	8.2k	OOM
6000	6.6k	OOM

(a) Computation speed (steps/hour)

	model	
batch_size	BASE	BIG
500	21.7M	11.9M
1000	30.2M	13.5M
1500	33.4M	14.7M
2000	33.7M	15.0M
2500	36.0M	16.2M
3000	37.0M	OOM
4500	36.7M	OOM
6000	39.4M	OOM

(b) Training throughput (subwords/hour)

Table 2: Computation speed and training throughput for a single GPU.

tion speed decreases with increasing batch size because not all operations in GPU are fully batch-parallelizable. The training throughput grows sub-linearly with increasing batch size, so based on these experiments only, there is just a small advantage when setting the batch size to the maximum value. We will return to this question in Section 4.5, while taking into account the translation quality.

We can also see the BASE model has approximately two times bigger throughput as well as computation speed relative to the BIG model.

GPUs	steps/hour	subwords/hour
1	9.8k	14.7M
2	7.4k	22.2M
6	5.4k	48.6M
8	5.6k	67.2M

Table 3: Computation speed and training throughput for various numbers of GPUs, with the BIG model and `batch_size=1500`.

Table 3 uses the BIG model and `batch_size=1500`, while varying the number of GPUs. The overhead in GPU synchronization is apparent from the decreasing computation speed. Nevertheless, the training throughput still grows with more GPUs, so e.g. with 6 GPUs we process 3.2 times more training data per hour relative to a single GPU (while without any overhead we would hypothetically expect 6 times more data).

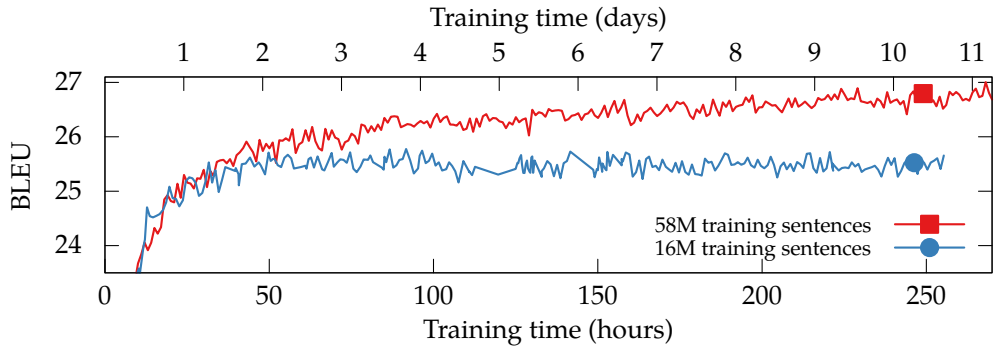


Figure 1: Training data size effect. BLEU learning curves for our main training dataset with 58 million sentence pairs and an alternative training dataset with 16 million sentence pairs. Both trained with 8 GPUs, BIG model and `batch_size=1500`.

The overhead when scaling to multiple GPUs is smaller than the overhead when scaling to a higher batch size. Scaling from a single GPU to 6 GPUs increases the throughput 3.2 times, but scaling from batch size 1000 to 6000 on a single GPU increases the throughput 1.3 times.

4.2. Training Data Size

For this experiment, we substituted CzEng 1.7 with CzEng 1.0 in the training data, so the total training size is 16 million sentence pairs (255M / 226M of English/Czech words). Figure 1 compares the BLEU learning curves of two experiments which differ only in the training data: the baseline CzEng 1.7 versus the smaller CzEng 1.0. Both are trained on the same hardware with the same hyper-parameters (8 GPUs, BIG, `batch_size=1500`). Training on the smaller dataset (2.5 times smaller in the number of words) converges to BLEU of about 25.5 after two days of training and does not improve over the next week of training. Training on the bigger dataset gives slightly worse results in the first eight hours of training (not shown in the graph) but clearly better results after two days of training, reaching over 26.5 BLEU after eight days.[18]

With `batch_size=1500` and 8 GPUs, training one epoch of the smaller dataset (with CzEng 1.0) takes 27k steps (5 hours of training), compared to 83k steps (15 hours) for the bigger dataset (with CzEng 1.7). This means *about 10 epochs in the smaller dataset were needed for reaching the convergence* and this is also the moment when the bigger

[18] We compared the two datasets also in another experiment with two GPUs, where CzEng 1.7 gave slightly worse results than CzEng 1.0 during the first two days of training but clearly better results after eight days. We hypothesize CzEng 1.0 is somewhat cleaner than CzEng 1.7.

dataset starts being clearly better. However, *even 18 epochs in the bigger dataset were not enough to reach the convergence*. *enough to reach the convergence*

Tips on Training Data Size

- For comparing different datasets (e.g. smaller and cleaner vs. bigger and noisier), we need to train long enough because *results after first hours (or days if training on a single GPU) may be misleading*.
- For large training data (as CzEng 1.7 which has over half a gigaword), *BLEU improves even after one week of training on eight GPUs* (or after 20 days of training on two GPUs in another experiment).
- *We cannot easily interpolate one dataset results to another dataset.* While the smaller training data (with CzEng 1.0) converged after 2 days, the main training data (with CzEng 1.7), which is 2.5 times bigger, continues improving even after 2.5×2 days.[19]

4.3. Model Size

Choosing the right model size is important for practical reasons: larger models may not fit any more on your GPU or they may require to use a very small batch size.

We experiment with two models,[20] as pre-defined in Tensor2Tensor – `transformer_big_single_gpu` (BIG) and `transformer_base_single_gpu` (BASE), which differ in four hyper-parameters summarized in Table 4.

model	hidden_size	filter_size	num_heads	adam_beta2
BASE	512	2048	8	0.980
BIG	1024	4096	16	0.998

Table 4: `transformer_big_single_gpu` (BIG) and `transformer_base_single_gpu` (BASE) hyper-parameter differences.

Figure 2 shows that on a single GPU, the BIG model becomes clearly better than the BASE model after 4 hours of training if we keep the batch size the same – 2000 (and we have confirmed it with 1500 in other experiments). However, the BASE model takes less memory, so we can afford a higher batch size, in our case 4500 (with no `max_length` restriction, see the next section), which improves the BLEU (see Section 4.5). But even

[19] Although such an expectation may seem naïve, we can find it in literature. For example, Bottou (2012) in Section 4.2 writes: "*Expect the validation performance to plateau after a number of epochs roughly comparable to the number of epochs needed to reach this point on the small training set.*"

[20] We tried also a model three times as large as BASE (1.5 times as large as BIG), but it did not reach better results than BIG, so we don't report it here.

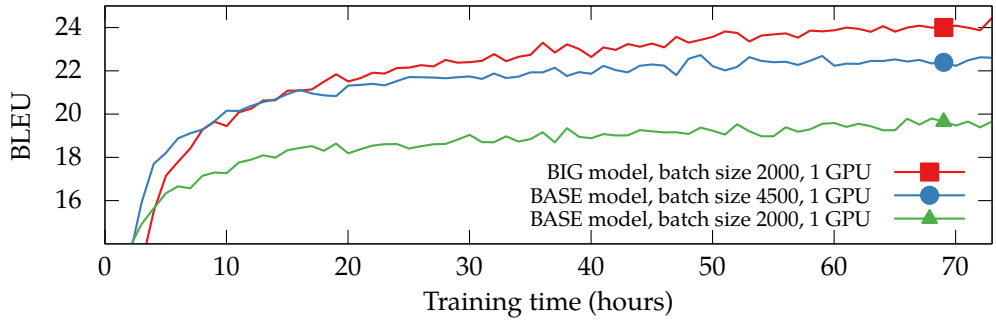


Figure 2: Effect of model size and batch size on a single GPU.

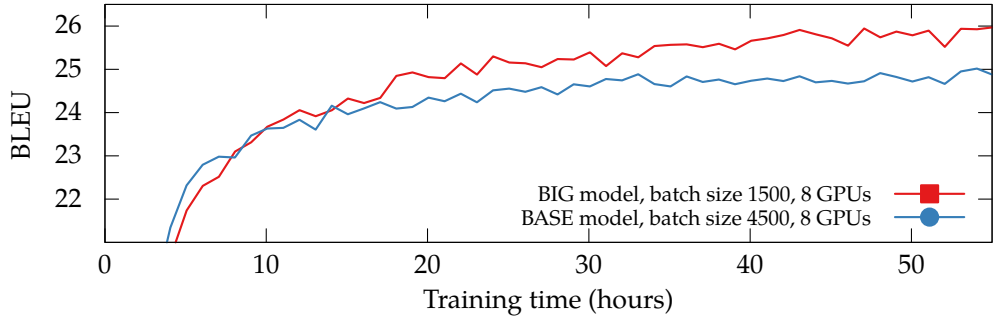


Figure 3: Effect of model size and batch size on 8 GPUs.

so, after less than one day of training, BIG with batch size 2000 becomes better than BASE with batch size 4500 (or even 6000 with `max_length=70` in another experiment) and the difference grows up to 1.8 BLEU after three days of training.

Figure 3 confirms this with 8 GPUs – here BIG with batch size 1500 becomes clearly better than BASE with batch size 4500 after 18 hours of training.

Tips on Model Size

- *Prefer the BIG over the BASE model* if you plan to train longer than one day and have 11 GB (or more) memory available on GPU.
- With less memory you should benchmark BIG and BASE with the maximum possible batch size.

	maximum batch size			longer sentences	
max_length	BIG+Adam	BIG+Adafactor	BASE+Adam	train	test
none	2040	2550	4950	0.0%	0.0%
150	2230	2970	5430	0.2%	0.0%
100	2390	3280	5990	0.7%	0.3%
70	2630	3590	6290	2.1%	2.2%
50	2750	3770	6430	5.0%	9.1%

Table 5: Maximum batch size which fits into 11GB memory for various combinations of `max_length` (maximum sentence length in subwords), model size (base or big) and optimizer (Adam or Adafactor). The last two columns show the percentage of sentences in the train (CzEng 1.7) and test (wmt13) data that are longer than a given threshold.

- For fast debugging (of model-size-unrelated aspects) use a model called `transformer_tiny`.

4.4. Maximum Training Sentence Length

The parameter `max_length` specifies the maximum length of a sentence in subwords. Longer sentences (either in source or target language) are excluded from the training completely. If no `max_length` is specified (which is the default), `batch_size` is used instead. Lowering the `max_length` allows to use a higher batch size or a bigger model. Since the Transformer implementation in T2T can suddenly run out of memory even after several hours of training, it is good to know how large batch size fits in your GPU. Table 5 presents what we empirically measured for the BASE and BIG models with Adam and Adafactor[21] optimizers and various `max_length` values.

Setting `max_length` too low would result in excluding too many training sentences and biasing the translation towards shorter sentences, which would hurt the translation quality. The last two columns in Table 5 show that setting `max_length` to 70 (resp. 100) results in excluding only 2.1% (resp. 0.7%) of sentences in the training data, and only 2.2% (resp. 0.3%) sentences in the development test data are longer, so the detrimental effect of smaller training data and length bias should be minimal in this setting. However, our experiments with `batch_size`=1500 in Figure 4 show a strange drop in BLEU after one hour of training for all experiments with `max_length` 70 or lower. Even with `max_length` 150 or 200 the BLEU learning curve is worse than with `max_length`=400, which finally gives the same result as not using any `max_length`

[21] The Adafactor optimizer (Shazeer and Stern, 2018) is available only in T2T 1.4.2 or newer and has three times smaller models than Adam because it does not store first and second moments for all weights. We leave further experiments with Adafactor for future work.

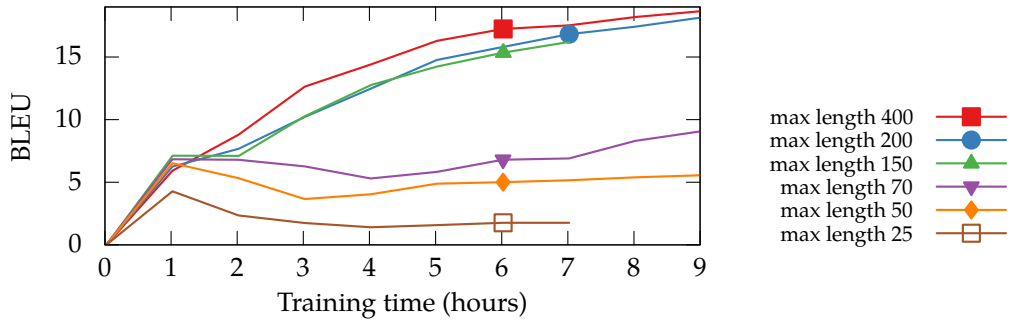


Figure 4: Effect of restricting the training data to various `max_length` values. All trained on a single GPU with the BIG model and `batch_size`=1500. An experiment without any `max_length` is not shown, but it has the same curve as `max_length`=400.

restriction. The training loss of `max_length`=25 (and 50 and 70) has high variance and stops improving after the first hour of training but shows no sudden increase (as in the case of diverged training discussed in Section 4.6 when the learning rate is too high). We have no explanation for this phenomenon.[22]

We did another set of experiments with varying `max_length`, but this time with `batch_size`=2000 instead of 1500. In this case, `max_length` 25 and 50 still results in slower growing BLEU curves, but 70 and higher has the same curve as no `max_length` restriction. So in our case, *if the batch size is high enough, the `max_length` has almost no effect on BLEU*, but this should be checked for each new dataset.

We trained several models with various `max_length` for three days and observed that *they are not able to produce longer translations than what was the maximum length used in training*, even if we change the decoding parameter alpha.

Tips on `max_length`

- *Set (a reasonably low) `max_length`.* This allows to use a higher batch size and prevents out-of-memory errors after several hours of training. Also, with a higher percentage of training sentences that are almost `max_length` long, there is a higher chance that the training will fail either immediately (if the batch size is too high) or never (otherwise).,
- *Set a reasonably high `max_length`.* Consider the percentage of sentences excluded from training and from the targeted development test set and also watch for unexpected drops (or stagnations) of the BLEU curve in the first hours of training.

[22] `https://github.com/tensorflow/tensor2tensor/issues/582`

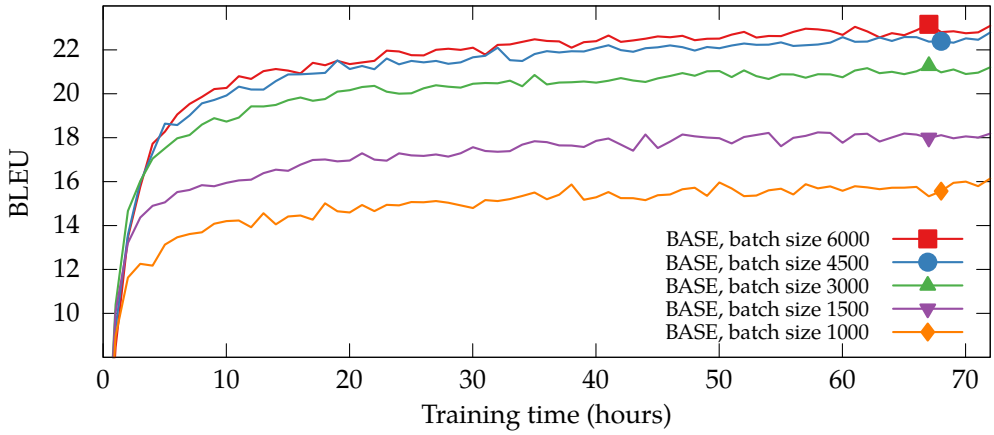


Figure 5: Effect of the batch size with the BASE model. All trained on a single GPU.

4.5. Batch Size

The default `batch_size` value in recent T2T versions is 4096 subwords for all models except for `transformer_base_single_gpu`, where the default is 2048. However, we recommend to always set the batch size explicitly[23] or at least make a note what was the default in a given T2T version when reporting experimental results.

Figure 5 shows learning curves for five different batch sizes (1000, 1500, 3000, 4500 and 6000) for experiments with a single GPU and the BASE model.[24] A higher batch size *up to 4500* is clearly better in terms of BLEU as measured by Time Till Score and Examples Till Score metrics defined in Section 4.1. For example, to get over BLEU of 18 with `batch_size`=3000, we need 7 hours (260M examples), and with `batch_size`=1500, we need about 3 days (2260M examples) i.e. 10 times longer (9 time more examples). From Table 2a we know that bigger batches have slower computation speed, so when re-plotting Figure 5 with steps instead of time on the x-axis, the difference between the curves would be even bigger. From Table 2b we know that bigger batches have slightly higher training throughput, so when re-plotting with number of examples processed on the x-axis, the difference will be smaller, but still visible. The only exception is the difference between batch size 4500 and 6000, which is very small and can be fully

[23] e.g. `--hparams="batch_size=1500,learning_rate=0.20,learning_rate_warmup_steps=16000"`
As the batch size is specified in subwords, we see no advantage in using power-of-two values.

[24] All the experiments in Figure 5 use `max_length`=70, but we have got the same curves when re-running without any `max_length` restrictions, except for `batch_size`=6000 which failed with OOM.

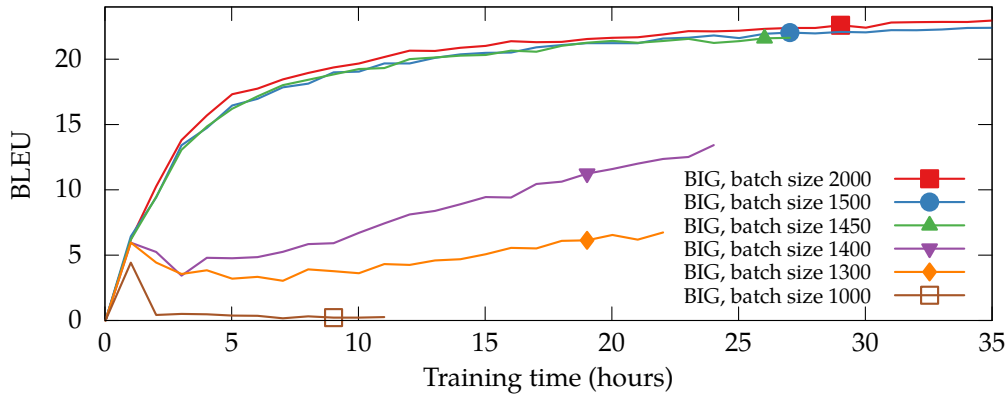


Figure 6: Effect of the batch size with the BIG model. All trained on a single GPU.

explained by the fact that batch size 6000 has 7% higher throughput than batch size 4500.

So *for the BASE model, a higher batch size gives better results*, although with diminishing returns. This observation goes against the common knowledge in other NMT frameworks and deep learning in general (Keskar et al., 2017) that smaller batches proceed slower (training examples per hour) but result in better generalization (higher test-set BLEU) in the end. In our experiments with the BASE model in T2T, bigger batches are not only faster in training throughput (as could be expected), but also faster in convergence speed, Time Till Score and Examples Till Score.

Interestingly, when replicating these experiments *with the BIG model, we see quite different results*, as shown in Figure 6. The BIG model needs a certain minimal batch size to start converging at all, but for higher batch sizes there is almost no difference in the BLEU curves (but still, bigger batch never makes the BLEU worse in our experiments). In our case, the sharp difference is between batch size 1450, which trains well, and 1400, which drops off after two hours of training, recovering only slowly.

According to Smith and Le (2017) and Smith et al. (2017), the *gradient noise scale*, i.e. scale of random fluctuations in the SGD (or Adam etc.) dynamics, is proportional to learning rate divided by the batch size (cf. Section 4.8). Thus when lowering the batch size, we increase the noise scale and the training may *diverge*. This may be either permanent, as in the case of batch size 1000 in Figure 6, or temporary, as in the case of batch size 1300 and 1400, where the BLEU continues to grow after the temporary drop, but much more slowly than the non-diverged curves.

We are not sure what causes the difference between the BASE and BIG models with regards to the sensitivity to batch size. One hypothesis is that the BIG model is more

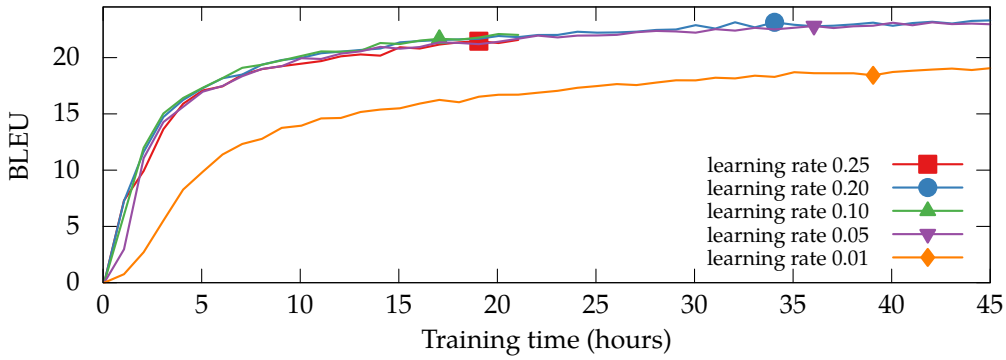


Figure 7: Effect of the learning rate on a single GPU. All trained on CzEng 1.0 with the default batch size (1500) and warmup steps (16k).

difficult to initialize and thus more sensitive to divergence in the early training phase. Also while for BASE, increasing the batch size was highly helpful until 4500, for BIG this limit may be below 1450, i.e. below the minimal batch size needed for preventing diverged training.

Tip on Batch Size

- *Batch size should be set as high as possible* while keeping a reserve for not hitting the out-of-memory errors. It is advisable to establish the largest possible batch size before starting the main and long training.

4.6. Learning Rate and Warmup Steps on a Single GPU

The default learning rate in T2T translation models is 0.20. Figure 7 shows that varying the value within range 0.05–0.25 makes almost no difference. Setting the learning rate too low (0.01) results in notably slower convergence. Setting the learning rate too high (0.30, not shown in the figure) results in *diverged* training, which means in this case that the learning curve starts growing as usual, but at one moment drops down almost to zero and stays there forever.

A common solution to prevent diverged training is to decrease the `learning_rate` parameter or increase `learning_rate_warmup_steps` or introduce gradient clipping. The `learning_rate_warmup_steps` parameter configures a `linear_warmup_rsqrt_decay` schedule[25] and it is set to 16 000 by default (for the BIG model), meaning that within

[25] The schedule was called `noam` in T2T versions older than 1.4.4.

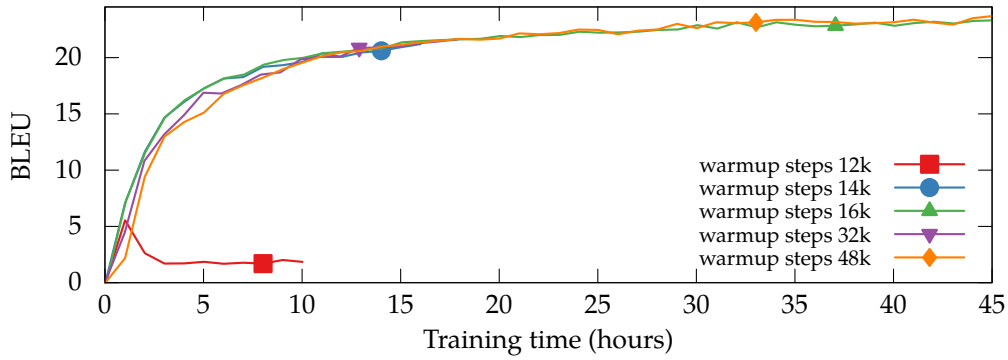


Figure 8: Effect of the warmup steps on a single GPU. All trained on CzEng 1.0 with the default batch size (1500) and learning rate (0.20).

the first 16k steps the learning rate grows linearly and then follows an inverse square root decay ($t^{-0.5}$, cf. Section 4.8.3). At 16k steps, the actual learning rate is thus the highest.

If a divergence is to happen, it usually happens within the first few hours of training, when the actual learning rate becomes the highest. Once we increased the warmup steps from 16k to 32k, we were able to train with the learning rate of 0.30 and even 0.50 without any divergence. The learning curves looked similarly to the baseline one (with default values of 16k warmup steps and learning rate 0.20). When trying learning rate 1.0, we had to increase warmup steps to 60k (with 40k the training diverged after one hour) – this resulted in a slower convergence at first (about 3 BLEU lower than the baseline after 8 hours of training), but after 3–4 days of training having the same curve as the baseline.

Figure 8 shows the effect of different warmup steps with a fixed learning rate (the default 0.20). Setting warmup steps too low (12k) results in diverged training. Setting them too high (48k, green curve) results in a slightly slower convergence at first, but matching the baseline after a few hours of training.

We can conclude that for a single GPU and the BIG model, there is a relatively large range of learning rate and warmup steps values that achieve the optimal results. The default values `learning_rate`=0.20 and `learning_rate_warmup_steps`=16000 are within this range.

Tips on Learning Rate and Warmup Steps

- *In case of diverged training, try gradient clipping and/or more warmup steps.*

- If that does not help (or if the warmup steps are too high relative to the expected total training steps), try decreasing the learning rate.
- Note that when you decrease warmup steps (and keep learning rate), you also increase the maximum actual learning rate because of the way how the `linear_warmup_rsqrt_decay` (aka `noam`) schedule is implemented.[26]

4.7. Number of GPUs

T2T allows to train with multiple GPUs on the same machine simply using the parameter `--worker_gpus`.[27] As explained in Section 2.3, the parameter `batch_size` is interpreted per GPU, so with 8 GPUs, the *effective batch size* is 8 times bigger.

A single-GPU experiment with batch size 4000, should give exactly the same results as two GPUs and batch size 2000 and as four GPUs and batch size 1000 because the effective batch size is 4000 in all three cases. We have confirmed this empirically. By the "same results" we mean BLEU (or train loss) versus training steps on the x-axis. When considering time, the four-GPU experiment will be the fastest one, as explained in Section 4.1.

Figure 9 shows BLEU curves for different numbers of GPUs and the BIG model with batch size, learning rate and warmup steps fixed on their default values (1500, 0.20 and 16k, respectively). As could be expected, training with more GPUs converges faster. What is interesting is the Time Till Score. Table 6 lists the approximate training time and number of training examples (in millions of subwords) needed to "surpass" (i.e. achieve and never again fall below) BLEU of 25.6.

# GPUs	hours	subwords (M)
1	> 600	> 9000
2	203	$2322 \cdot 2 = 4644$
6	56	$451 \cdot 6 = 2706$
8	40	$341 \cdot 8 = 2728$

Table 6: Time and training data consumed to reach BLEU of 25.6, i.e. Time Till Score and Examples Till Score. Note that the experiment on 1 GPU was ended after 25 days of training without clearly surpassing the threshold (already outside of Figure 9).

[26] This holds at least in T2T versions 1.2.9–1.5.2, but as it is somewhat unexpected/unintuitive for some users, it may be fixed in future, see `https://github.com/tensorflow/tensor2tensor/issues/517`.

[27] and making sure environment variable `CUDA_VISIBLE_DEVICES` is set so enough cards are visible. T2T allows also distributed training (on multiple machines), but we have not experimented with it. Both single-machine multi-gpu and distributed training use synchronous Adam updates by default.

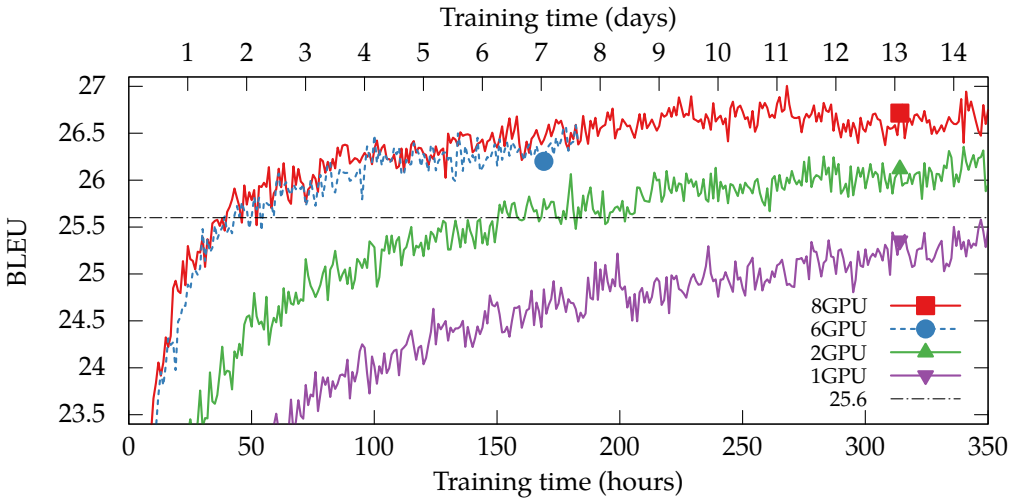


Figure 9: Effect of the number of GPUs. BLEU=25.6 is marked with a black line.

We can see that *two GPUs are more than three times faster than a single GPU* when measuring the Time Till Score and need much less training examples (i.e. they have lower Examples Till Score). Similarly, *eight GPUs are more than five times faster than two GPUs* and 1.7 times less training data is needed.

Recall that in Figure 6 we have shown that increasing the batch size from 1450 to 2000 has almost no effect on the BLEU curve. However, when increasing the effective batch size by using more GPUs, the improvement is higher than could be expected from the higher throughput.[28] We find this quite surprising, especially considering the fact that we have not tuned the learning rate and warmup steps (see the next section).

Tips on the Number of GPUs

- For the fastest BLEU convergence *use as many GPUs as available* (in our experiments up to 8).
- This holds *even when there are more experiments* to be done. For example, it is better to run one 8-GPUs experiment after another, rather than running two 4-GPUs experiments in parallel or eight single-GPU experiments in parallel.

[28] It would be interesting to try simulating multi-GPU training on a single GPU, simply by doing the update once after N batches (and summing the gradients). This is similar to the *ghost batches* of Hoffer et al. (2017), but using ghost batch size higher than the actual batch size. We leave this for future work.

4.8. Learning Rate and Warmup Steps on Multiple GPUs

4.8.1. Related Work

There is a growing number of papers on scaling deep learning to multiple machines with synchronous SGD (or its variants) by increasing the effective batch size. We will focus mostly on the question how to adapt the learning rate schedule, when scaling from one GPU (or any device, in general) to k GPUs.

Krizhevsky (2014) says "*Theory suggests that when multiplying the batch size by* k*, one should multiply the learning rate by* $\sqrt{k}$ *to keep the variance in the gradient expectation constant*", without actually explaining which theory suggests so. However, in the experimental part he reports that what worked the best, was a *linear scaling heuristics*, i.e. multiplying the learning rate by k, again without any explanation nor details on the difference between $\sqrt{k}$ scaling and k scaling.

The linear scaling heuristics become popular, leading to good scaling results in practice (Goyal et al., 2017; Smith et al., 2017) and also theoretical explanations (Bottou et al., 2016; Smith and Le, 2017; Jastrzebski et al., 2017). Smith and Le (2017) interpret SGD (and its variants) as a stochastic differential equation and show that the *gradient noise scale* $g = \epsilon\left(\frac{N}{B} - 1\right)$, where ϵ is the learning rate, N is the training set size, and B is the effective batch size. This noise "*drives SGD away from sharp minima, and therefore there is an optimal batch size which maximizes the test set accuracy*". In other words for keeping the optimal level of gradient noise (which leads to "flat minima" that generalize well), we need to scale the learning rate linearly when increasing the effective batch size.

However, Hoffer et al. (2017) suggest to use $\sqrt{k}$ scaling instead of the linear scaling and provide both theoretical and empirical support for this claim. They show that $\mathrm{cov}(\Delta w, \Delta w) \propto \frac{\epsilon^2}{NB}$, thus if we want to keep the the covariance matrix of the parameters update step Δw in the same range for any effective batch size B, we need to scale the learning rate proportionally to the square root of B. They found that $\sqrt{k}$ scaling works better than linear scaling on CIFAR10.[29] You et al. (2017) confirm linear scaling does not perform well on ImageNet and suggest to use Layer-wise Adaptive Rate Scaling.

We can see that large-batch training is still an open research question. Most of the papers cited above have experimental support only from the image recognition tasks (usually ImageNet) and convolutional networks (e.g. ResNet), so it is not clear whether their suggestions can be applied also on sequence-to-sequence tasks (NMT) with self-attentional networks (Transformer). There are several other differences as well: Modern convolutional networks are usually trained with *batch normalization*

[29] To close the gap between small-batch training and large-batch training, Hoffer et al. (2017) introduce (in addition to $\sqrt{k}$ scaling) so-called *ghost batch normalization* and *adapted training regime*, which means decaying the learning rate after a given number of steps instead of epochs.

(Ioffe and Szegedy, 2015), which seems to be important for the scaling, while Transformer uses *layer normalization* (Lei Ba et al., 2016).[30] Also, Transformer uses Adam together with an inverse-square-root learning-rate decay, while most ImageNet papers use SGD with momentum and piecewise-constant learning-rate decay.

4.8.2. Our Experiments

We decided to find out empirically the optimal learning rate for training on 8 GPUs. Increasing the learning rate from 0.20 to 0.30 resulted in diverged training (BLEU dropped to almost 0 after two hours of training). Similarly to our single-GPU experiments (Section 4.6), we were able prevent the divergence by increasing the warmup steps or by introducing gradient clipping (e.g. with `clip_grad_norm=1.0`, we were able to use learning rate 0.40, but increasing it further to 0.60 led to divergence anyway). However, *none of these experiments led to any improvements over the default learning rate* – all had about the same BLEU curve after few hours of training.

Jastrzebski et al. (2017) shows that "*the invariance under simultaneous rescaling of learning rate and batch size breaks down if the learning rate gets too large or the batch size gets too small*". A similar observation was reported e.g. by Bottou et al. (2016). Thus our initial hypothesis was that 0.20 (or 0.25) is the maximal learning rate suitable for stable training in our experiments even when we scale from a single GPU to 8 GPUs. Considering this initial hypothesis, we were surprised that we were able to achieve so good Time Till Score with 8 GPUs (more than 8 times smaller relative to a single GPU, as reported in Table 6). To answer this riddle we need to understand how learning rate schedules are implemented in T2T.

4.8.3. Parametrization of Learning Rate Schedules in T2T

In most works on learning rate schedules[31] the "time" parameter is actually interpreted as the number of epochs or training examples. For example a popular setup for piecewise-constant decay in ImageNet training (e.g. Goyal et al., 2017) is to divide the learning rate by a factor of 10 at the 30-th, 60-th, and 80-th epoch.

However, in T2T, it is the `global_step` variable that is used as the "time" parameter. So when increasing the effective batch size 8 times, e.g. by using 8 GPUs instead of a single GPU, the actual learning rate[32] achieves a given value after the same number of

[30] Applying batch normalization on RNN is difficult. Transformer does not use RNN, but still we were not successful in switching to batch normalization (and possibly ghost batch normalization) due to NaN loss errors.

[31] Examples of learning rate schedules are inverse-square-root decay, inverse-time decay, exponential decay, piecewise-constant decay, see `https://www.tensorflow.org/api_guides/python/train#Decaying_the_learning_rate` for TF implementations.

[32] By *actual* learning rate we mean the learning rate after applying the decay schedule. The `learning_rate` parameter stays the same in this case.

steps, but this means after 8 times less training examples. For the inverse-square-root decay, we have $actual\_lr(steps) = c \cdot steps^{-0.5} = \frac{1}{\sqrt{8}} \cdot actual\_lr(steps \cdot 8)$, where c is a constant containing also the `learning_rate` parameter. So with 8 GPUs, if we divide the `learning_rate` parameter by $\sqrt{8}$, we achieve the same actual learning rate after a given number of training examples as in the original single-GPU setting.

This explains the riddle from the previous section. *By keeping the `learning_rate` parameter the same when scaling to* k *times bigger effective batch, we actually increase the actual learning rate* $\sqrt{k}$ *times*, in accordance with the suggestion of Hoffer et al. (2017).[33] This holds only for the `linear_warmup_rsqrt_decay` (aka `noam`) schedule and ignoring the warmup steps.

If we want to keep the same learning rate also in the warmup phase, we would need to divide the warmup steps by k. However, this means that the maximum actual learning rate will be $\sqrt{k}$ times higher, relative to the single-GPU maximal actual learning rate and this leads to divergence in our experiments. In deed, many researchers (e.g. Goyal et al., 2017) suggest to use a warmup when scaling to more GPUs in order to prevent divergence. Transformer uses learning rate warmup by default even for single-GPU training (cf. Section 4.6), but it makes sense to use more warmup training examples in multi-GPU setting.

In our experiments with 8 GPUs and the default learning rate 0.20, using 8k warmup steps instead of the default 16k had no effect on the BLEU curve (it was a bit higher in the first few hours, but the same afterwards). Further decreasing the warmup steps resulted in a retarded BLEU curve (for 6k) or a complete divergence (for 2k).

Tips on Learning Rate and Warmup Steps on Multiple GPUs

- Keep the `learning_rate` parameter at its optimal value found in single-GPU experiments.
- You can try decreasing the warmup steps, but less than linearly and you should not expect to improve the final BLEU this way.

4.9. Resumed Training

T2T allows to resume training from a checkpoint, simply by pointing the `output_dir` parameter to a directory with an existing checkpoint (specified in the `checkpoint` file). This may be useful when the training fails (e.g. because of hardware error), when we need to continue training on a different machine or during hyper-parameter search, when we want to continue with the most promising setups. T2T saves also Adam

[33] In addition to suggesting the $\sqrt{k}$ learning-rate scaling, Hoffer et al. (2017) show that to fully close the "generalization gap", we need to train longer because the absolute number of steps (updates) matters. So from this point of view, using steps instead of epochs as the time parameter for learning rate schedules may not be a completely wrong idea.

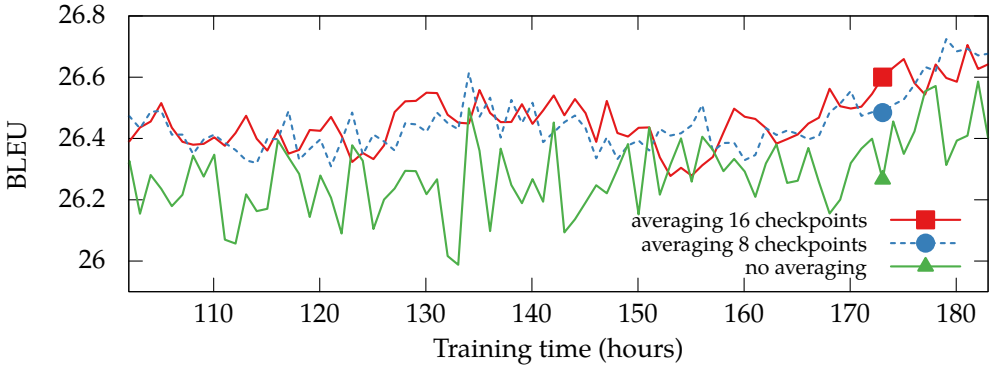


Figure 10: Effect of checkpoint averaging. All trained on 6 GPUs.

momentum into the checkpoint, so the training continues almost as if it had not been stopped. However, it does not store the position in the training data – it starts from a random position. Also the relative time (and wall-clock time) in TensorBoard graphs will be influenced by the stopping.

Resumed training can also be exploited for changing some hyper-parameters, which cannot be meta-parametrized by the number of steps. For example, Smith et al. (2017) suggest to increase the effective batch size (and number of GPUs) during training, instead of decaying the learning rate.

Yet another usage is to do domain adaptation by switching from (large) general-domain training data to (small) target-domain training data for the few last epochs. In this case, consider editing also the learning rate or learning rate schedule (or faking the `global_step` stored in the checkpoint) to make sure the learning rate is not too small.

4.10. Checkpoint Averaging

Vaswani et al. (2017) suggest to average the last 20 checkpoints saved in 10-minute intervals (using `utils/avg_checkpoints.py`). According to our experiments slightly better results are achieved with averaging checkpoints saved in 1-hour intervals. This has also the advantage that less time is spent with checkpoint saving, so the training is faster.

Figure 10 shows the effect of averaging is twofold: the averaged curve has lower variance (flickering) from checkpoint to checkpoint and it is almost always better than the baseline without averaging (usually by about 0.2 BLEU). In some setups, we have seen improvements due to averaging over 1 BLEU. In the early phases of training, while the (baseline) learning curve grows fast, it is better to use fewer checkpoints for

	Manual		Automatic Scores				
#	Ave %	Ave z	BLEU	TER	CharacTER	BEER	System
–	–	–	**23.8**	**0.662**	**0.582**	**0.543**	T2T 8 GPUs 8 days
1	**62.0**	**0.308**	*22.8*	*0.667*	*0.588*	*0.540*	uedin-nmt
2	*59.7*	*0.240*	20.1	0.703	0.612	0.519	online-B
3	55.9	0.111	20.2	0.696	0.607	0.524	limsi-factored
	55.2	0.102	20.0	0.699	-	-	LIUM-FNMT
	55.2	0.090	20.2	0.701	*0.605*	0.522	LIUM-NMT
	54.1	0.050	20.5	0.696	0.624	0.523	CU-Chimera
	53.3	0.029	16.6	0.743	0.637	0.503	online-A
8	41.9	-0.327	16.2	0.757	0.697	0.485	PJATK

Table 7: WMT17 systems for English-to-Czech and our best T2T training run. Manual scores are from the official WMT17 ranking. Automatic metrics were provided by `http://matrix.statmt.org/`. For *TER metrics, lower is better. Best results in bold, second-best in italics.

averaging. In later phases (as shown in Figure 10, after 4.5–7.5 days of training), it seems that 16 checkpoints (covering last 16 hours) give slightly better results on average than 8 checkpoints, but we have not done any proper evaluation for significance (using paired bootstrap testing for each hour and then summarizing the results).

The fact that resumed training starts from a random position in the training data (cf. Section 4.9) can be actually exploited for "forking" a training to get two (or more) copies of the model, which are trained for the same number of steps, but independently in the later stages and thus ending with different weights saved in the final checkpoint. These semi-independent models can be averaged in the same way as checkpoints from the same run, as described above. Our preliminary results show this helps a bit (on top of checkpoint averaging).

Tips on Checkpoint Averaging

- Use it. Averaging 8 checkpoints takes about 5 minutes, so it is a "BLEU boost for free" (compared with the time needed for the whole training).
- See the tools for automatic checkpoint averaging and evaluation described in Section 2.4.

5. Comparison with WMT17 Systems

Table 7 provides the results of WMT17 English-to-Czech news translation task, with our best Transformer model (BIG trained on 8 GPUs for 8 days, averaging 8 checkpoints) evaluated using the exact same implementation of automatic metrics. While the automatic evaluation is not fully reliable (see e.g. the high BLEU score for CU-Chimera despite its lower manual rank), we see that the Transformer model out-

performs the best system in BLEU, TER, CharacTER and BEER, despite it does not use any back-translated data, reranking with other models (e.g. right-to-left reranking) nor ensembling (as is the case of uedin-nmt and other systems). Note that our Transformer uses a subset of the constrained training data for WMT17, so the results are comparable.

6. Conclusion

We presented a broad range of basic experiments with the Transformer model (Vaswani et al., 2017) for English-to-Czech neural machine translation. While we limit our exploration to the more or less basic parameter settings, we believe this report can be useful for other researchers. In sum, experiments done for this article took about 4 years of GPU time.

Among other practical observations, we've seen that for the Transformer model, larger batch sizes lead not only to faster training but more importantly better translation quality. Given at least a day and a 11GB GPU for training, the larger setup (BIG) should be always preferred. The Transformer model and its implementation in Tensor2Tensor is also best fit for "intense training": using as many GPUs as possible and running experiments one after another should be preferred over running several single-GPU experiments concurrently.

The best performing model we obtained on 8 GPUs trained for 8 days has outperformed the WMT17 winner in a number of automatic metrics.

Acknowledgements

This research was supported by the grants 18-24210S of the Czech Science Foundation, H2020-ICT-2014-1-645452 (QT21) of the EU, SVV 260 453, and using language resources distributed by the LINDAT/CLARIN project of the Ministry of Education, Youth and Sports of the Czech Republic (LM2015071).

Bibliography

Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio. Neural Machine Translation by Jointly Learning to Align and Translate. In *Proceedings of ICLR*, 2015.

Bojar, Ondřej, Zdeněk Žabokrtský, Ondřej Dušek, Petra Galuščáková, Martin Majliš, David Mareček, Jiří Maršík, Michal Novák, Martin Popel, and Aleš Tamchyna. The Joy of Parallelism with CzEng 1.0. In *Proceedings of the Eighth International Language Resources and Evaluation Conference (LREC'12)*, pages 3921–3928, Istanbul, Turkey, May 2012. ELRA, European Language Resources Association. ISBN 978-2-9517408-7-7.

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 Sojka, Petr, Aleš Horák, Ivan Kopeček, and Karel Pala,

editors, *Text, Speech, and Dialogue: 19th International Conference, TSD 2016*, number 9924 in Lecture Notes in Artificial Intelligence, pages 231–238. Masaryk University, Springer International Publishing, 2016. ISBN 978-3-319-45509-9.

Bojar, Ondřej, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Matthias Huck, Philipp Koehn, Varvara Logacheva, Christof Monz, Matteo Negri, Matt Post, Raphael Rubino, Lucia Specia, and Marco Turchi. Findings of the 2017 Conference on Machine Translation (WMT17). In *Proceedings of the Second Conference on Machine Translation*, Copenhagen, Denmark, September 2017a. ACL.

Bojar, Ondřej, Yvette Graham, and Amir Kamran. Results of the WMT17 Metrics Shared Task. In *Proceedings of the Second Conference on Machine Translation*, Copenhagen, Denmark, September 2017b. ACL.

Bottou, Léon. *Stochastic Gradient Descent Tricks*, pages 421–436. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012. ISBN 978-3-642-35289-8. doi: 10.1007/978-3-642-35289-8_25. URL https://doi.org/10.1007/978-3-642-35289-8_25.

Bottou, L., F. E. Curtis, and J. Nocedal. Optimization Methods for Large-Scale Machine Learning. *ArXiv e-prints*, June 2016. URL https://arxiv.org/abs/1606.04838.

Cettolo, Mauro, Marcello Federico, Luisa Bentivogli, Jan Niehues, Sebastian Stüker, Katsuhito Sudoh, Koichiro Yoshino, and Christian Federmann. Overview of the IWSLT 2017 Evaluation Campaign. In *Proceedings of the 14th International Workshop on Spoken Language Translation (IWSLT)*, pages 2–14, Tokyo, Japan, 2017.

Goyal, Priya, Piotr Dollár, Ross B. Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, Large Minibatch SGD: Training ImageNet in 1 Hour. *CoRR*, 2017. URL http://arxiv.org/abs/1706.02677.

Hoffer, Elad, Itay Hubara, and Daniel Soudry. Train longer, generalize better: closing the generalization gap in large batch training of neural networks. 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 1731–1741. Curran Associates, Inc., 2017. URL http://papers.nips.cc/paper/6770-train-longer-generalize-better-closing-the-generalization-gap-in-large-batch-training-of-neural-networks.pdf.

Ioffe, Sergey and Christian Szegedy. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. *CoRR*, abs/1502.03167, 2015. URL http://arxiv.org/abs/1502.03167.

Jastrzebski, Stanislaw, Zachary Kenton, Devansh Arpit, Nicolas Ballas, Asja Fischer, Yoshua Bengio, and Amos J. Storkey. Three Factors Influencing Minima in SGD. *CoRR*, abs/1711.04623, 2017. URL http://arxiv.org/abs/1711.04623.

Keskar, Nitish Shirish, Dheevatsa Mudigere, Jorge Nocedal, Mikhail Smelyanskiy, and Ping Tak Peter Tang. On Large-Batch Training for Deep Learning: Generalization Gap and Sharp Minima. In *Proceedings of ICLR*, 2017. URL http://arxiv.org/abs/1609.04836.

Krizhevsky, Alex. One weird trick for parallelizing convolutional neural networks. *CoRR*, abs/1404.5997, 2014. URL http://arxiv.org/abs/1404.5997.

Lee, Jason, Kyunghyun Cho, and Thomas Hofmann. Fully Character-Level Neural Machine Translation without Explicit Segmentation. *CoRR*, 2016. URL http://arxiv.org/abs/1610.03017.

Lei Ba, J., J. R. Kiros, and G. E. Hinton. Layer Normalization. *ArXiv e-prints*, July 2016.

Papineni, Kishore, Salim Roukos, Todd Ward, and Wei-Jing Zhu. BLEU: a Method for Automatic Evaluation of Machine Translation. In *Proceedings of ACL 2002*, pages 311–318, Philadelphia, Pennsylvania, 2002.

Popović, Maja. chrF: character n-gram F-score for automatic MT evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395, Lisbon, Portugal, September 2015. ACL. URL http://aclweb.org/anthology/W15-3049.

Sennrich, Rico, Barry Haddow, and Alexandra Birch. Neural Machine Translation of Rare Words with Subword Units. In *Proceedings of ACL 2016*, pages 1715–1725, Berlin, Germany, August 2016. ACL. URL http://www.aclweb.org/anthology/P16-1162.

Shazeer, N. and M. Stern. Adafactor: Adaptive Learning Rates with Sublinear Memory Cost. *ArXiv e-prints*, Apr. 2018. URL https://arxiv.org/abs/1804.04235.

Smith, Samuel L. and Quoc V. Le. A Bayesian Perspective on Generalization and Stochastic Gradient Descent. In *Proceedings of Second workshop on Bayesian Deep Learning (NIPS 2017)*, Long Beach, CA, USA, 2017. URL http://arxiv.org/abs/1710.06451.

Smith, Samuel L., Pieter-Jan Kindermans, and Quoc V. Le. Don't Decay the Learning Rate, Increase the Batch Size. *CoRR*, 2017. URL http://arxiv.org/abs/1711.00489.

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz 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 6000–6010. Curran Associates, Inc., 2017. URL http://papers.nips.cc/paper/7181-attention-is-all-you-need.pdf.

Wu, Yonghui, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Lukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation. *CoRR*, abs/1609.08144, 2016. URL http://arxiv.org/abs/1609.08144.

You, Yang, Igor Gitman, and Boris Ginsburg. Scaling SGD Batch Size to 32K for ImageNet Training. *CoRR*, abs/1708.03888, 2017. URL http://arxiv.org/abs/1708.03888.

Address for correspondence:
Martin Popel
popel@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 110 APRIL 2018 71–84

Search for the Relation of Form and Function Using the ForFun Database

Marie Mikulová, Eduard Bejček, Eva Hajičová, Jarmila Panevová

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

Abstract

The aim of the contribution is to introduce a database of linguistic forms and their functions built with the use of the multi-layer annotated corpora of Czech, the Prague Dependency Treebanks. The purpose of the Prague Database of Forms and Functions (ForFun) is to help the linguists to study the form-function relation, which we assume to be one of the principal tasks of both theoretical linguistics and natural language processing. We demonstrate possibilities of the exploitation of the ForFun database.

This article is largely based on a paper presented at the 16th International Workshop on Treebanks and Linguistic Theories in Prague (Bejček et al., 2017).

1. Introduction

The study of the relation between (linguistic) forms and their functions or meanings (terms known from Saussure's structural linguistics (Saussure, 1916) as the relation between "signifié" and "signifiant") is one of the fundamental tasks of linguistics, with important implications for natural language understanding. As Katz (1966, p. 100) says, to understand the ability of natural languages to serve as an instrument to the communication of thoughts and ideas we must understand what it is that permits those who speak them consistently to connect the right sounds with the right meanings. This, however, is obviously not an easy task as the relation between form and function is a many-to-many relation. At present, the availability of richly annotated corpora helps the linguist to analyze the given relation in its variety, and it is a challenging task to provide linguists with useful tools for their study.

 Corresponding author: mikulova@ufal.mff.cuni.cz
Cite as: Marie Mikulová, Eduard Bejček, Eva Hajičová, Jarmila Panevová. Search for the Relation of Form and Function Using the ForFun Database. The Prague Bulletin of Mathematical Linguistics No. 110, 2018, pp. 71–84. doi: 10.2478/pralin-2018-0003.

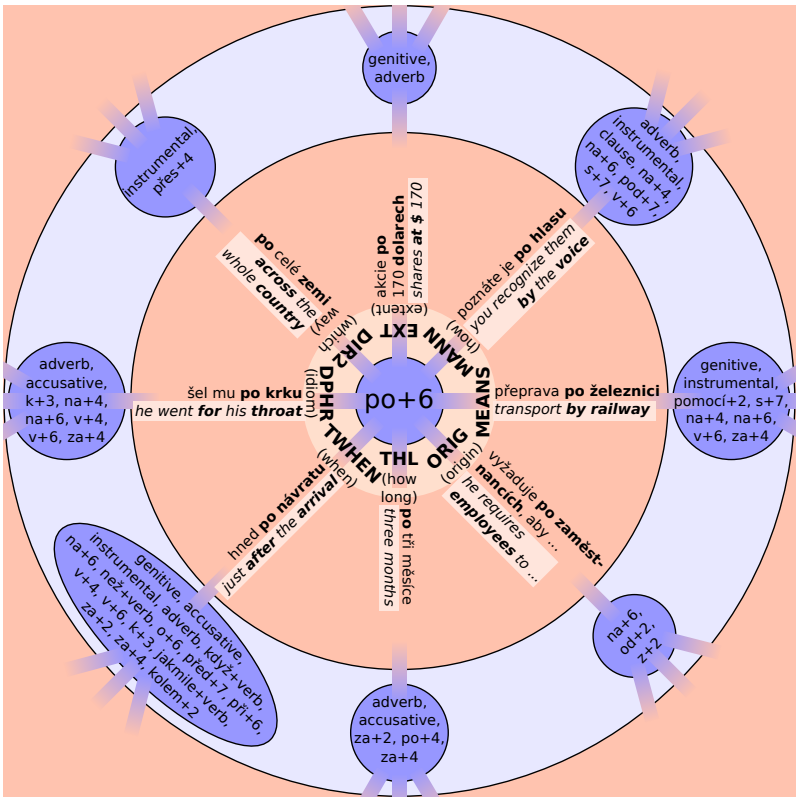


Figure 1. Many-to-many relation between forms and functions demonstrated on prepositional case po + Locative.

One of the most useful types of corpora for this task are treebanks based on a stratificational (multi-layer) approach, where the form-function relation may be understood as a relation between units of two layers of the system. The aim of this paper is to introduce a database of language forms and their linguistic functions built with the use of the multi-layer annotated corpora of Czech, the Prague Dependency Treebanks (PDTs), with the purpose to help the linguists to study the form-function relation. We offer a new database ForFun which gives a possibility to search in a user-friendly way all forms (almost 1 500 items) used in PDTs for particular functions and vice versa to look up all functions (66 items) expressed by the particular forms.

The research question we follow by constructing the database can be illustrated e.g. by the example of the Czech preposition *po* + Locative case of a noun (translated to English as *along*, *on*, *about*, *at*, … + noun) in Figure 1. The dark colour indicates the

forms, the light colour the functions, identified in the PDTs by the functors attached to the nodes representing the given item (see below Section 2).[1] The prepositional case *po* + Locative (see the inner circle) may express the following eight functions (see the middle circle): TWHEN (when), THL (how long), ORIG (origin), MEANS, MANN (manner), EXT (extent), DIR2 (direction which way), DPHR (idiomatic meaning). Each of these functions, in turn, may be expressed by a number of forms (see the outer circle) one of which is *po* + Locative. Thus for example, the function labelled THL (how long) may be expressed by an adverb, or Accusative of a noun (prepositionless case), or prepositional cases *za* + Genitive, *za* + Accusative, *po* + Accusative, and, of course, by the already mentioned *po* + Locative. In Figure 1, only a few functions of *po* + Locative are displayed; for a full list of 32 functions see their list in Table 3.

2. Multi-layer Architecture of Prague Dependency Treebanks

PDTs (on which our ForFun database is based) are complex linguistically motivated treebanks based on the dependency syntactic theory of the Functional Generative Description (see Sgall et al. 1986). The original annotation scheme has the following multi-layer architecture:[2]

- **morphological layer**: all tokens of the sentence get a lemma and a (disambiguated) morphological tag,
- **surface syntax layer** (analytical): a dependency tree capturing surface syntactic relations such as subject, object, adverbial; a (structural) tag reflecting these relations is attached to the nodes as one component of their (complex) labels,
- **deep syntax layer** (tectogrammatical) capturing the semantico-syntactic relations: on this layer, the dependency structure of a sentence is a tree consisting of nodes only for autonomous meaningful units (function words such as prepositions, subordinating conjunctions, auxiliary verbs etc. are not represented as separate nodes in the structure, their contribution to the meaning of the sentence is captured within the complex labels of the autonomous units). The types of dependency relations are captured by means of the so-called functors.

Functors (66 in total) are classified according to different criteria. The basic subdivision is based on the the valency criterion, which divides functors into the argument functors and adjunct functors. There are five arguments: Actor/Bearer (ACT), Patient (PAT), Addressee (ADDR), Origin (ORIG) and Effect (EFF). The repertory of adjuncts is

[1] Throughout the paper, we use the term *functor* for the label of the type of the dependency relation between the governor and its dependent; in the dependency tree structure representing the sentence on the deep (underlying, tectogrammatical; see Section 2) layer this label is a part of the complex label attached to the dependent node. The term *prepositional case* is used for a combination of a preposition and a noun or a nominal group in a morphological case. In the figures and tables, morphological cases are indicated by numbers, i.e. 2 for Genitive, 3 for Dative, 4 for Accusative, 6 for Locative, 7 for Instrumental. When the noun or nominal group is not accompanied by a preposition, we use the term *prepositionless case*.

[2] The PDTs annotation scenario is described in detail in Mikulová et al. (2006) and Hajič et al. (2017).

much larger than that of arguments. Their set might be divided into several subclasses, such as temporal (TWHEN for "when?", TSIN for "since when?", TTILL for "till when?", TPAR for "during what time?", THL for "how long?", THO for "how often?", TFHL for "for how long?", TFRWH for "from when?", and TOWH for "to when?"), local (LOC for "where?", DIR1 for "where from?", DIR2 for "which way?", DIR3 for "where to?"), causal (CAUS for "cause", AIM for "aim", INTT for "intention", COND for "condition", CNCS for "concession"), functors for manner (MANN for general "manner", MEANS for "means or instrument"), and other functors for other adjuncts (such as ACMP for "accompaniment", EXT for "extent", INTF for "intensifier", BEN for "benefactor", etc.). For a full list of all dependency relations and their labels see Mikulová et al. (2006).

The nodes on a lower layer are explicitly referenced from the corresponding closest (immediately higher) layer. These links allow for tracing every unit of annotation all the way down to the original raw text. For the ForFun database, we use the annotations of the nodes on the deep syntactic layer and their counterparts on the morphological layer, which has made it possible to retrieve the relations between functions (expressed on the deep layer by functors) and forms and vice versa.

3. List of available Prague Dependency Treebanks

For Czech, the following four treebanks are available, each of them contains data of a different source. The Prague Dependency Treebank version 3.5 (PDT 3.5),[3] the newest edition of the core Prague Dependency Treebank, consists of articles from Czech daily newspapers. A slightly modified scenario was used for the annotation of the Prague Czech-English Dependency Treebank 2.0 (PCEDT 2.0),[4] the Prague Dependency Treebank of Spoken Czech 2.0 (PDTSC 2.0),[5] and the PDT-Faust corpus. In contrast to the original PDT project, in these treebanks, the morphological and surface syntactic annotations were done automatically, and the manually annotated deep syntactic layer does not contain some special annotations. However, the annotation of functors, which is important for our research of the form-function relation, has been done manually in all treebanks.

In the parallel PCEDT 2.0 (Hajič et al., 2012), the English part consists of the Wall Street Journal sections of the Penn Treebank (Marcus et al., 1993), and the Czech part, which is used in the ForFun database, was manually translated from the English original. PDTSC 2.0 (Mikulová et al., 2017b) contains dialogs from the Malach project[6] (slightly moderated testimonies of Holocaust survivors) and from the Companions

[3] https://ufal.mff.cuni.cz/pdt3.5

[4] https://ufal.mff.cuni.cz/pcedt2.0/

[5] https://ufal.mff.cuni.cz/pdtsc2.0

[6] https://ufal.mff.cuni.cz/cvhm/vha-info.html

project[7] (two participants chat over a collection of photographs). PDT-Faust is a small treebank containing short segments (very often with vulgar content) typed in by various users on the reverso.net webpage for translation.

It is obvious (see Table 1) that the Prague Dependency Treebank family provides rich language data for our purpose, i.e. for the study of the relation of forms and their functions since every content word there is assigned one of those 66 functors. Altogether, the treebanks contain around 180 000 sentences with their morphological, syntactic and semantic annotation.

	PDT 3.0	PCEDT 2.0	PDTSC 2.0	Faust	Total
Tokens	833 195	1 162 072	742 257	33 772	2 771 296
Sentences	49 431	49 208	73 835	3 000	175 474

Table 1. Volume of data in Prague Depencency Treebanks

4. Prague Database of Forms and Functions

ForFun 1.0, the Prague Database of Forms and Functions (Mikulová and Bejček, 2018), is a rich database of syntactic functions and their formal realizations with a large amount of examples coming from both written and spoken Czech texts. Since the database is extracted from the PDTs (see Section 3), it takes over the list of syntactic functions as well as the terminology (they are called *functors*).

ForFun is provided as a digital open source accessible to all scholars via the LINDAT/CLARIN repository.[8]

4.1. Design

We have already mentioned that in general the relation between forms and functions is a many-to-many relation. As such, it has to be explored from both sides: a given form has several functions and any of these functions may again be realized by several forms (the given one among them). When such relations have to be explored, ForFun is a perfect choice, since it is designed exactly for this kind of traversing through data.

Although the annotated example sentences are the same, they can be retrieved by asking either for their forms or for their functions. The ForFun database provides two entry points (cf. Figures 2 and 3):

[7] http://cordis.europa.eu/project/rcn/96289_en.html

[8] http://hdl.handle.net/11234/1-2542

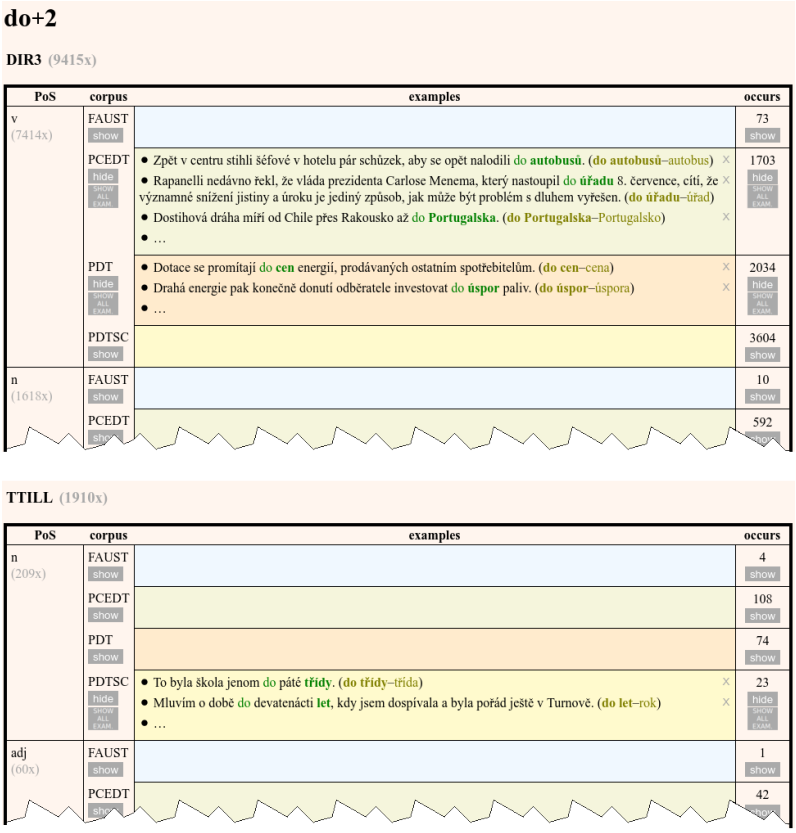


Figure 2. A screenshot of the ForFun web interface: From Form to Function.

- The user can choose one of almost 1 500 formal realizations of sentence units (i.e. prepositionless and prepositional cases, subordinated and coordinate conjunctions, adverbs, infinitive and finite verb forms, etc.) and obtains all functions it can represent.
- The user can choose one of 66 syntactic functions (i.e. `LOC`, `TTILL`, `CAUS` etc.) and obtains all forms used to express it.

The view can be always switched from a list of forms to a list of functions of one of them and vice versa.

For each form-function relation there are plenty of examples in the form of a sentence with the highlighted expression representing the relation. All these examples are sorted by various criteria:

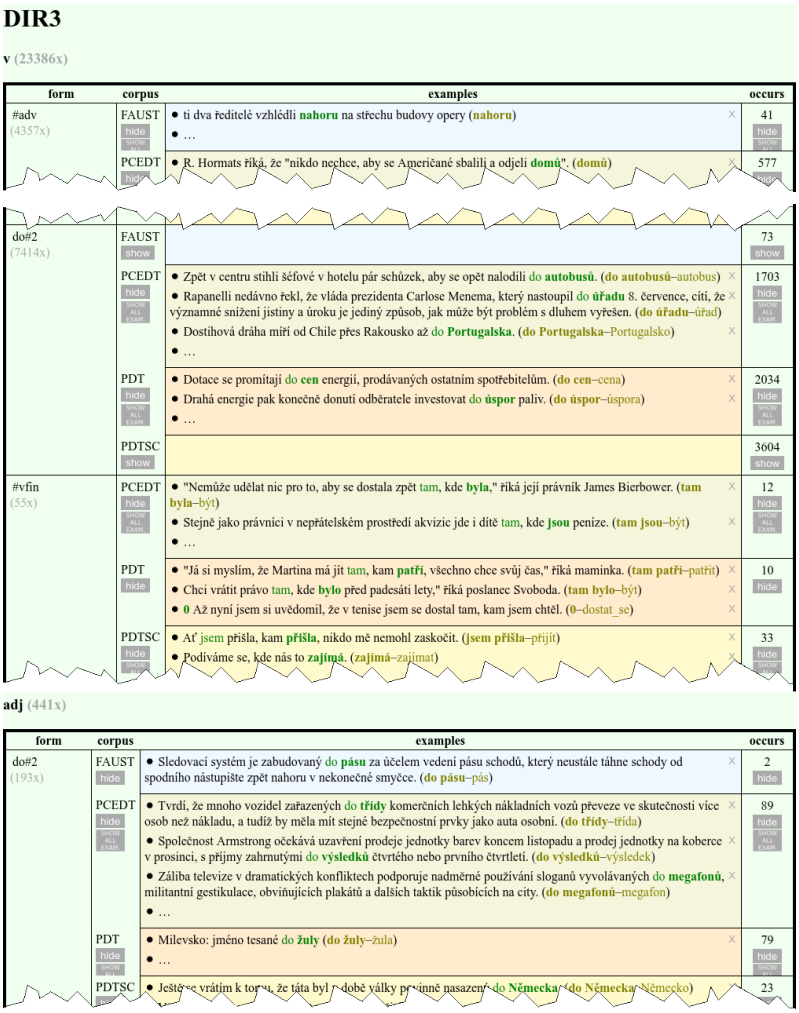


Figure 3. A screenshot of the ForFun web interface: From Function to Form.

- the word class of the parent node,
- the particular forms for the function or particular functions for the form, and
- the source of data (written, spoken, translated texts and texts from internet).

The number of examples available in the database is displayed for each pair form + functor, or functor + word class, each combination functor + form + word class and each specified 4-combination (form + functor + word class + source). Either first ten

examples or all of them are displayed on demand. On top of that, examples can be also first filtered by their source, which allows the user to hide e.g. all forms used only in spoken language or use only sentences from written corpora.

An illustration of how the result of user's search for the functions of the prepositional case *do* + Genitive looks like is given in Figure 2. In the upper part of the screenshot of the ForFun web interface, there are 9 415 occurrences in all PDTs of the form *do* + Genitive representing the functor `DIR3`. The occurrences of *do* + Genitive are divided according to their heads (be it a v(erb) or a n(oun), see the first column); their distribution within particular treebank is given in the second column followed by real examples from the corresponding treebank. A few of them are displayed on demand whereas many (see the last column) stay hidden. In the lower part of Figure 2, the same form *do* + Genitive in the function `TTILL` is exemplified in the same style. Note that Figure 2 presents only a part of the full response obtained from the ForFun database for the given query. The other functions of *do* + Genitive (`PAT`, `EXT`, `EFF` and others) are also not included in this shortened sample. (The list of all functions expressed by *do* + Genitive is in Table 3.)

For the opposite direction "from function to form" see the screenshot in Figure 3, where (among others) the same sentences for *do* + Genitive as the functor `DIR3` can be found searching for all representations of the functor `DIR3`. Other forms include a finite verb (#vfin) or an adverb (#adv).

4.2. Volume

The ForFun database contains 2.2 million examples altogether for all forms (and the same number from the function point of view), split approx. 3:1 between written and spoken text (see Table 2). Each example is one sentence long.[9] They can be examined from the function side (66 functors) or the form side (1 469 forms). All examples are split into 13.5 thousand of 4-combinations (form + functor + word class + source), each with 163 examples in average.

While the average number is high, median is only two examples. The reason is that there is a long tail of 4-combinations used very rarely. These occurences with very low frequencies in the data are one of the main benefits of the large volume of database, but they have to be used carefully. Every result has to be always understood solely as an input for a subsequent research, as ForFun may contain errors (caused by annotators as well as speakers/writers) considering its volume.

[9] One sentence typically contains many different functions and serves for many examples (once for each of its parts).

examples from written text	1 608 061
examples from spoken text	593 400
examples altogether	2 201 461
number of functions	66
number of forms	1 469
number of 4-combinations	13 514
avg. examples for a function	33 355
avg. examples for a form	1 500
avg. examples for a 4-combination	163
max. number of examples for a function	490 121
max. number of examples for a form	370 586
max. number of examples for a 4-combination	97 469

Table 2. Volume of the ForFun database

5. Possibilities of the Exploitation of the ForFun Database

To display the richness of the material we work with, we present several examples connected with the studies of the form-function relation what the user can find out in the ForFun database.

5.1. Multi-functionality of Forms

A rather straightforward use of the ForFun database is to retrieve which functions can be expressed by the particular form and which forms can express the particular function. Table 3 contains seven prepositional cases with the highest number of functions they express: *na* + Accusative, *v* + Locative, *k* + Dative, *za* + Accusative, *do* + Genitive, and *po* + Locative (cf. Figure 1).

5.2. Functions with the Most Limited List of Forms

Table 4, by contrast with Table 3, displays those functions that are expressed by the smallest number of forms (not only prepositional cases, but also other possible forms). We can observe that the HER (heritage), CONTRD contradiction, and TFRWH (from-when) functions are expressed exclusively by a single form. E.g. functor HER (heritage) is expressed exclusively by the form *po* + Locative, but HER belongs to many functions (32 in total) which are expressed by *po* + Locative (cf. their list in Table 3).

prep.	number	list of functors
na+4	42	ACT ADDR AIM APP ATT BEN CAUS COMPL COND CPHR CPR CRIT DIFF DIR1 DIR3 DPHR EFF EXT ID INTF INTT LOC MANN MAT MEANS MOD ORIG PAT PREC REG RESL RESTR RHEM RSTR SUBS TFHL TFRWH THL TOWH TPAR TTILL TWHEN
v+6	36	ACMP ACT AIM APP ATT BEN CAUS COMPL COND CPR CRIT DENOM DIR2 DIR3 DPHR EFF EXT ID LOC MANN MAT MEANS MOD PAT PREC REG RESL RESTR RHEM RSTR SUBS TFHL THL THO TPAR TWHEN
k+3	34	ACMP ACT ADDR AIM APP ATT BEN CAUS COMPL CPHR CRIT DIR1 DIR2 DIR3 DPHR EFF EXT ID INTT LOC MANN PAR PAT PREC REG RESL RESTR RHEM RSTR TOWH TPAR TSIN TTILL TWHEN
za+4	33	ACMP ACT AIM APP BEN CAUS CNCS COMPL COND CPHR DIR1 DIR3 DPHR EFF EXT HER ID INTT LOC MANN MEANS ORIG PAT PREC REG RSTR SUBS TFHL TFRWH THL THO TPAR TWHEN
na+6	33	ACT ADDR AIM APP ATT BEN CAUS COND CPR CRIT DIR2 DIR3 DPHR EFF EXT ID INTT LOC MANN MEANS ORIG PAR PAT PREC REG RESL RESTR RSTR TFHL THO TOWH TPAR TWHEN
do+2	33	ADDR AIM APP ATT BEN COMPL COND CPHR DIR1 DIR3 DPHR EFF EXT INTT LOC MANN MEANS MOD OPER PAR PARTL PAT REG RESL RSTR TFHL THL THO TOWH TPAR TSIN TTILL TWHEN
po+6	32	ACT AIM APP CAUS COND CPR CRIT DIR2 DIR3 DPHR EXT HER ID INTT LOC MANN MAT MEANS ORIG PAR PAT REG RSTR SUBS TFHL THL THO TOWH TPAR TSIN TTILL TWHEN

Table 3. The prepositional cases with the highest number of functions.

5.3. Absolute Frequency of Forms and Functions (in both written and spoken texts)

An observation of frequency has an important place in the description of language because it quantifies linguistic choices made by speakers and writers. For each form and function, ForFun provides information about raw frequency in all PDTs as well as in each corpus separately. The users can search quickly and in a user-friendly way which formal means are the most frequent in Czech sentences and which ones are rarely used. See Table 5 for five most frequent prepositional cases in comparison with the class of adverbs and the clause with the conjunction *že* 'that'.

The users of ForFun can also find out the distribution of a particular function (various arguments or adjuncts) in the sentences. For both forms and functions, they can compare their absolute frequencies in written and spoken texts. In Table 6, the sub-

functor	meaning	list of forms	example
HER	heritage	***po+6***	*Podědila tu nemoc* ***po rodičích****.* 'She inherited the disease **from parents**.'
CONTRD	contradiction	***zatímco+verb***	*On byl jedináček,* ***zatímco*** *ona měla dvanáct dětí.* 'He was an only child, **while** she had twelve children.'
TFRWH	from when	***z+2***	***Ze*** *kterého* ***roku*** *je tato fotka?* '**From** which **year** is this photo?'
TOWH	to when	***na+4***; *pro+4*	*Derby je vypsáno* ***na*** *3.* ***září****.* 'Derby is listed **on September** 3.'
TSIN	since when	***od+2***; *z+2*; *adverb*	*V energetice pracuje* ***od roku*** *1964.* 'He has worked in energetics **since 1964**.'
THO	how often	***adverb***; *Acc*; *Instr*	*Pořádáte přechod* ***každý rok?*** 'Do you organize march **every year**?'
TTILL	till when	*do+2*; *dokud+verb*; ***adverb***; *než+verb*	*Smlouva nebyla* ***dodnes*** *podepsána.* 'No contract has been signed **yet**.'

Table 4. Functions with the most limited list of forms.

classification of the most frequent functors for adjuncts is presented in comparison of their presence in written and spoken texts. We see that spatial and temporal functors (see their list in Section 2) are by far the most frequently occurring adjunct types. Hypothetically, in a Czech text of 100 sentences, there would be 61 sentences containing an adjunct (or several different adjuncts) and out of these sentences there would be: 29 sentences with spatial functor(s), 26 with temporal functor(s), 12 with manner functor(s), 10 with causal functor(s) and 22 with other functor(s).

5.4. Material for Detailed Linguistic Studies

In addition to valuable statistical data, the ForFun database provides an extremely rich material for detailed linguistic studies of individual language phenomena and for their description and classification, e.g., valency behavior, coordination/discourse relations, idioms and complex predicates, comparison of written and spoken texts, etc. The first linguistic studies based on the database analyze and subclassify the functors denoting space and time (Mikulová et al., 2017a, 2018). The studies perform a detailed description of subtle meanings of temporal and spatial adjuncts including a list of formal means with real examples coming from both written and spoken texts and as such demonstrate that ForFun can be used for fundamental linguistic research.

form	occurences
v+6	51 682
na+4	22 444
s+7	19 747
z+2	19 502
na+6	17 870
adverb	93 824
že[that]+verb	26 831

Table 5. The most frequent prepositional cases.

sentences containing:	all texts	%	written texts	spoken texts
spatial functors	74 164	29	43 089	31 075
temporal functors	66 503	26	42 266	24 237
functors for manner	31 583	12	21 752	9 831
causal functors	26 569	10	18 022	8 547
other functors for adjuncts	50 425	20	35 967	14 458
no functor for adjuncts	99 564	39	60 060	39 504

Table 6. The frequency distribution of the selected group of functors

6. Conclusion

The ForFun database has been built as a rich and user-friendly resource for those researchers who (want to) use corpora in their everyday work and look for various occurrences of specific forms or patterns in relation to their syntactic functions etc. but they are not interested or just do not need to deal with various technical, formal and annotation issues. ForFun brings a rich and complex annotation in PDTs based on a sound linguistic theory closer to common researchers. It will be further developed, though it should be borne in mind that it is designed to provide only a limited number of most useful features, rather than a full interface to everything PDTs can offer. There are other complex tools for that[10] and ForFun does not aim to substitute them. In its simplicity and clarity, it is a user-friendly source of examples for various explorations especially in syntax.

[10] E.g. PML Tree Query `https://lindat.mff.cuni.cz/services/pmltq/`, INESS Search `http://clarino.uib.no/iness`, etc.

Acknowledgments

This article is largely based on a paper presented at the 16th International Workshop on Treebanks and Linguistic Theories in Prague (Bejček et al., 2017).

The research reported in the paper has been supported by the Czech Science Foundation under the projects GA17-12624S and GA17-07313S and by the LINDAT/CLARIN project of Ministry of Education, Youth and Sports of the Czech Republic (project LM2015071). This work has been using language resources developed, stored and distributed by the latter project (LM2015071).

Bibliography

Bejček, Eduard, Eva Hajičová, Marie Mikulová, and Jarmila Panevová. The Relation of Form and Function in Linguistic Theory and in a Multilayer Treebank. In *Proceedings of the 16th International Workshop on Treebanks and Linguistic Theories*, pages 56–63, 2017. URL `http://aclweb.org/anthology/W17-7609`.

Hajič, Jan, Eva Hajičová, Jarmila Panevová, Petr Sgall, Ondřej Bojar, Silvie Cinková, Eva Fučíková, 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 *Proceedings of the 8th International Conference on Language Resources and Evaluation (LREC 2012)*, pages 3153–3160. European Language Resources Association, Istanbul, Turkey, 2012. ISBN 978-2-9517408-7-7. URL `https://aclanthology.info/pdf/L/L12/L12-1280.pdf`.

Hajič, Jan, Eva Hajičová, Marie Mikulová, and Jiří Mírovský. *Handbook on Linguistic Annotation*, chapter Prague Dependency Treebank, pages 555–594. Springer Verlag, Dordrecht, Netherlands, 2017.

Katz, Jerrold J. *The philosophy of language*. Studies in languages. Harper & Row, New York, 1966.

Marcus, Mitchell P., Beatrice Santorini, and Mary Ann Marcinkiewicz. Building a Large Annotated Corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330, 1993. URL `https://aclanthology.info/pdf/J/J93/J93-2004.pdf`.

Mikulová, Marie and Eduard Bejček. ForFun 1.0: Prague Database of Syntactic Forms and Functions – An Invaluable Resource for Linguistic Research. In *Proceedings of the 11th International Conference on Language Resources and Evaluation (LREC 2018)*, Paris, France, 2018. European Language Resources Association.

Mikulová, Marie, Alevtina Bémová, Jan Hajič, Eva Hajičová, Jiří Havelka, Veronika Kolářová, Lucie Kučová, Markéta Lopatková, Petr Pajas, Jarmila Panevová, Magda Razímová, Petr Sgall, Jan Štěpánek, Zdeňka Urešová, Kateřina Veselá, and Zdeněk Žabokrtský. Annotation on the tectogrammatical level in the Prague Dependency Treebank. Annotation manual. Technical Report 30, ÚFAL MFF UK, Prague, Czech Rep., 2006.

Mikulová, Marie, Eduard Bejček, Veronika Kolářová, and Jarmila Panevová. Subcategorization of Adverbial Meanings Based On Corpus Data. *Journal of Linguistics / Jazykovedný časopis*, 68(2):268–277, 2017a. ISSN 0021-5597.

Mikulová, Marie, Jiří Mírovský, Anna Nedoluzhko, Petr Pajas, Jan Štěpánek, and Jan Hajič. PDTSC 2.0 – Spoken Corpus with Rich Multi-layer Structural Annotation. In *Text, Speech, and Dialogue 20th International Conference, TSD 2017*, Lecture Notes in Computer Science, pages 129–137, Cham / Heidelberg / New York / Dordrecht / London, 2017b. Charles University, Springer International Publishing. ISBN 978-3-319-64206-2.

Mikulová, Marie, Eduard Bejček, and Jarmila Panevová. What Can We Find Out about Time and Space in the ForFun Database? In *Proceedings of the Second Workshop on Corpus-Based Research in the Humanities CRH-2*, Gerastree proceedings, pages 133–142. Austrian Academy of Science, Dept. of Geoinformation, Wien, Austria, 2018. ISBN 978-3-901716-43-0.

Saussure, Ferdinand de. *Cours de linguistique générale*. C. Bally and A. Sechehaye, with the collaboration of A. Riedlinger, eds. Lausanne and Paris: Payot, 1916.

Sgall, Petr, Eva Hajičová, and Jarmila Panevová. *The Meaning of the Sentence and Its Semantic and Pragmatic Aspects*. Academia/Reidel Publishing Company, Prague/Dordrecht, 1986.

Address for correspondence:
Marie Mikulová
mikulova@ufal.mff.cuni.cz
Charles University, Faculty of Mathematics and Physics,
Institute of Formal and Applied Linguistics
Malostranské náměstí 25, 118 00 Prague 1, Czech Republic

PBML

The Prague Bulletin of Mathematical Linguistics
NUMBER 110 APRIL 2018 85–101

Improving Topic Coherence Using Entity Extraction Denoising

Ronald Cardenas,[a] Kevin Bello,[a] Alberto Coronado,[a] Elizabeth Villota[b]

[a] Department of Mechanical Engineering, National University of Engineering, Lima, Peru
[b] Section of Mechanical Engineering, Pontifical Catholic University of Peru, Lima, Peru

Abstract

Managing large collections of documents is an important problem for many areas of science, industry, and culture. Probabilistic topic modeling offers a promising solution. Topic modeling is an unsupervised machine learning method and the evaluation of this model is an interesting problem on its own. Topic interpretability measures have been developed in recent years as a more natural option for topic quality evaluation, emulating human perception of coherence with word sets correlation scores. In this paper, we show experimental evidence of the improvement of topic coherence score by restricting the training corpus to that of relevant information in the document obtained by Entity Recognition. We experiment with job advertisement data and find that with this approach topic models improve interpretability in about 40 percentage points on average. Our analysis reveals as well that using the extracted text chunks, some redundant topics are joined while others are split into more *skill-specific* topics. Fine-grained topics observed in models using the whole text are preserved.

1. Introduction

Probabilistic topic models, such as Latent Dirichlet Allocation (Blei et al., 2003) and its many variants (Newman et al., 2006; Blei and Lafferty, 2005, 2006; Teh et al., 2006; Blei et al., 2007), were introduced in an unsupervised setting to discover latent semantic structures in a collection of documents, namely the topics. However, there is no guarantee that the inferred topics – typically modeled as a set of important words – are easily interpretable by humans.

Traditionally, held-out likelihood had been used to perform topic model evaluation. Chang et al. (2009) conducted a study that showed that perplexity actually corre-

 Corresponding author: racardenasa@uni.pe
Cite as: Ronald Cardenas, Kevin Bello, Alberto Coronado, Elizabeth Villota. Improving Topic Coherence Using Entity Extraction Denoising. The Prague Bulletin of Mathematical Linguistics No. 110, 2018, pp. 85–101.
doi: 10.2478/pralin-2018-0004.

lates negatively with human interpretability of such topics. In other words, choosing the model with the lowest perplexity on unseen data may generate topics that are hardly interpretable. This motivates the search of different evaluation methods for topic modeling, referred in the literature as topic coherence measures (Newman et al., 2010; Musat et al., 2011; Mimno et al., 2011; Stevens et al., 2012; Aletras and Stevenson, 2013; Lau et al., 2014).

In this work, we hypothesize that topic interpretability – as measured by topic coherence – can be improved by training a topic model over text chunks of relevant information instead of the whole text per document, for job advertisement posts published in job-hunting websites. We analyze two scenarios of how categories of skills required for a specific job vacancy span across professional majors. The first scenario is a noisy scenario in which the topics are inferred using all the information available in job ads which includes e.g. company description, payment, working schedule. In the second scenario, the topics are inferred only over specific information about the job itself, such as expected skills, tasks to perform, and professional major of preference, extracted by named entity recognition. We find that this last setup scenario successfully increases coherence scores of inferred topics, obtains much cleaner topics and is able to infer meaningful clusters of majors related by the skills applicants are required to know.

This article is structured as follows. We first present related work on the field. Then, in section 3 we present all the theoretical background necessary to formulate the problem tackled. In section 4, the experimental setup of every module is thoroughly explained, and the dataset used in presented as well. Section 5 presents the results and discussion of our findings. Finally, section 6 presents the conclusions.

2. Related Work

In recent years, several topic coherence measures have been proposed (Newman et al., 2010; Musat et al., 2011; Mimno et al., 2011; Stevens et al., 2012; Aletras and Stevenson, 2013; Lau et al., 2014) in order to automate the method of Chang et al. (2009) and emulate human interpretability. Newman et al. (2010) introduced the notion of coherence and was the first to propose an automatic measure based on pairwise pointwise mutual information (PMI) between the topic words. Subsequent empirical works on topic coherence proposed measures based on word statistics that differ in several details, such as normalization (Lau et al., 2014), aggregation methods (Mimno et al., 2011), and reference corpus (Musat et al., 2011; Aletras and Stevenson, 2013). Röder et al. (2015) proposed a framework for the exploration of all possible coherence measures, modeled as a pipeline where the blocks (e.g. aggregation method, confirmation measure) can be exchanged and create new measures. They combined two complementary lines of research on coherence: scientific coherence and topic modeling.

As the acceptance of topic coherence measures increases as a mean of topic model assessment (Paul and Girju, 2010; Reisinger et al., 2010; Hall et al., 2012), recent research trends focus on proposing fast and efficient models that can be scaled up to big amounts of data (Yang et al., 2015; Nguyen et al., 2015), using the whole text per document for training.

Prior to directly evaluating human interpretability, several approaches were proposed to improve topic quality. Airoldi et al. (2010) analyzed the effect of varying the source text and inference strategies for PNAS biological sciences publications, obtaining a slightly higher number of new categories that better explain nowadays intertwined research fields. The usage of name entities as extra information in a topic model is explored by Newman et al. (2006). They propose a customized probabilistic graphical model that directly learns the entity-topic relationship and making better predictions about entities.

3. Problem Formulation

We define the problem of improving topic coherence as follows. Given a collection of highly noisy documents, we extract only relevant information from each document in the form of custom entities. The extraction task is modeled as a sequence labeling problem, and we tackle it by using the averaged structured perceptron (see Section 3.1).

As test case, we consider the domain of job advertisements. A job ad contains valuable information about what skills applicants are expected to have, but they contain spurious information as well. In order to avoid inferring topics over noise, we extract requirements, functions and preferred major from a job ad using a custom named entity recognition and extraction pipeline.

We now present notation and definitions core to the modules our model is based. We start by formally defining the entity extractor module, followed by the topic modeling. Then, the coherence metric is presented.

3.1. Averaged Structured Perceptron

The structured perceptron and its averaged version was initially introduced by Collins (2002). They differ from the well-known perceptron algorithm in that the output for each training instance pair $(x_t, y_t) \in T$ is a structure $y' \in Y_t$, where Y_t is the space of permissible structured outputs for input x. The inference algorithm to predict y' is problem dependent. In our case, sequence labeling, a first order *Viterbi* decoder is used. In each step, the candidate y' is transformed to a high-dimensional feature representation $f(x, y) \in R^m$ and the prediction is determined by a linear classifier based on the dot product of this representation and a weight vector $w \in R^m$.

In practice, this algorithm can be implemented easily and behaves remarkably well in several problems. These two characteristics make the structured perceptron algorithm a natural first choice for prototyping structured models.

3.2. Latent Dirichlet Allocation

In this section, we briefly describe the graphical model called Latent Dirichlet Allocation (LDA) (Blei et al., 2003), originally proposed for doing topic modeling. LDA is a generative probabilistic model in which the data is in the form of a collection of documents, and each document in the form of a collection of words. The model assumes that each document is a mixture of latent topics, and each topic is modeled as a mixture of words. These random mixture distributions are considered Dirichlet-distributed to be inferred from the data. The generative process of LDA can be described as follow:

1. For all D documents sample $\theta_d \sim \mathrm{Dir}(\alpha)$.
2. For all K topics sample $\phi_k \sim \mathrm{Dir}(\beta)$.
3. For each of the N_d words v_i in document d:
   - Sample a topic $z_i \sim \mathrm{Multinomial}(\theta_d)$
   - Sample a word $v_i \sim \mathrm{Multinomial}(\phi_{z_i})$
   - Observe the word

We assume symmetric Dirichlet priors for θ_d and ϕ_k, as suggested by Griffiths and Steyvers (2004).

Regarding inference strategies for the models, we make use of Gibbs Sampling as described in Griffiths and Steyvers (2004) and the Variational Expectation - Maximization (VEM) algorithm as described in Blei et al. (2003).

3.3. Topic Coherence

We use the coherence metric proposed by Mimno et al. (2011), based in conditional log likelihood of co-occurrence of top topic word pairs. We refer to it as *UMass* coherence from now on. It is defined as follows:

$$C_{UMass} = \frac{2}{N \cdot (N-1)} \sum_{i=2}^{N} \sum_{j=1}^{i-1} \log \frac{P(w_i, w_j) + \epsilon}{P(w_j)},$$

where N is the number of top words in a topic to consider.

4. Experimental Setup

4.1. Job Ads Corpus

The job ads corpus (Cardenas Acosta et al., 2016) was built by extracting job ads from several popular job search websites in Peru, and it is divided in two parts, one for entity extraction tasks and the other for topic inference.

The first part consists of 400,000 word tokens spanning 800 posts manually labeled with entity tags following the CoNLL-2000 BIO tagging format (Ramshaw and Marcus, 1995). This amount of data proved to give good results for named entity extraction in Spanish, as reported by Carreras et al. (2002). The custom entities defined for our task are FUN (tasks to be performed at the job), REQ (skills required) and CARR (preferred professional major of the applicant). Table 1 show an example of annotation along with its translation into English, whereas Table 2 shows the proportion of entities in the annotated corpus as well as the average length in words.

Spa	Egresado/*O* en/*O* Ingeniería/*B-CARR* de/*I-CARR* Software/*I-CARR* con/*O* conocimientos/*O* de/*O* base/*B-REQ* de/*I-REQ* datos/*I-REQ* MySQL/*I-REQ* .
Eng	Graduate in Software Engineering with knowledge of MySQL databases

Table 1: Example of tagging of custom entities

Entity	**Number of chunks**	**Avg. number of words per chunk**
FUN	3291	11.09
REQ	4833	1.84
CARR	2097	1.64

Table 2: Defined entities and presence in corpus

The second part consists of only job ads requesting engineering professions published between January and March 2015. We compose each document instance as the concatenation of the title and description fields of each job ad. We consider 23 engineering categories and leave out categories with less than 50 posts. Since the same job ad can be published in more than one website, we consider it as repeated if the same description of the position is found within the last fifteen days in the database. The final topic inference corpus consists of 9,472 job ads, with an average of 91.3 ± 40.8 tokens per document and a total of 476,990 tokens.

The dataset is publicly available in the Lindat repository.[1]

[1] http://hdl.handle.net/11234/1-2673

4.2. Data Preparation

Job ads often contain very sparse information like emails, dates, office hours and salary. We treated this type of tokens as noise and replaced them with appropriate tags (e.g. URL) using regular expressions. Low-frequency words were filtered as well, following Bikel et al. (1999) approach of using generic labels based on orthographic features (e.g. Capitalized, hasDigit, AllCaps).

4.3. Skills and Tasks Extraction

We train one tagger for each entity, each one with the following features. Note that each feature is conditioned to the current label being predicted, unless otherwise specified (e.g. transition features).

- Trigger word features for the current word (Carreras et al., 2002), only for REQ and CARR entities.
- Lowercase form and position of all words in a window of $\pm n$ words (Carreras et al., 2002). For the CARR entity, $n = 2$ and for the others $n = 3$.
- Stemmed form and position of previous, current and next word.
- Part-of-list feature ($\mathtt{list} :: y_i$), if current word is part of a list.
- Orthographic features, including long-word and single-digit (Carreras et al., 2002), for previous, current and next word.
- Suffix and prefix features, last and first 3 characters respectively, for previous, current and next word.
- Word brown-cluster mapping features (Miller et al., 2004) for previous, current and next word.
- Token bigram and trigram emission features (Liang and Collins, 2005) for lowercase and stemmed form of all words, as well as orthographic class, in a window of ± 2 words.
- Relative position of sentence in document, if the current sentence belongs to the document border (first one or last two sentences). Only used for FUN entity.
- Bigram transition features for word cluster mapping (Liang and Collins, 2005), used only for REQ entity.
- Bigram transition features (Liang and Collins, 2005) for lowercase and stemmed form, as well as orthographic class, of each word in the bigram.
- Bigram transition features of last states (labels) predicted.

Preliminary experiments showed that POS information does not contribute significantly to the taggers' performance. Additionally, usage of a Conditional Random Field model (Lafferty et al., 2001) showed no significant improvements with respect to the Averaged Perceptron. We also considered using pre-trained word embeddings as input, but the limited amount of data available would not allow us to obtain reliable estimates. On the other hand, pre-training the embeddings on a large monolingual benchmark and then training over our data would not allow the model to learn ter-

minology not only specific to the domain but to the Spanish dialect spoken in the country in which the ads where published.

The annotated dataset is divided in 70, 15 and 15 percent for training, validation and testing, respectively. The evaluation metrics are the standard precision P (fraction of output chunks that exactly match the reference chunks), recall R (fraction of reference chunks returned by the tagger), and their harmonic mean, the F_1 score, $F_1 = 2 \times P \times R/(P + R)$. The accuracy rate for individual labeling decisions is overoptimistic as an accuracy measure for NER, given that O labels are more frequent. Even so, we report BIO accuracy for reference.

4.4. Topic Modeling

We employ the analysis approach suggested by Airoldi et al. (2010), aimed to explore the effect of varying the data source over model dimensionality and using different hyperparameters inference strategies and algorithms (Variational Inferences vs Gibbs sampling).

We explore models both estimating and fixing the latent categories proportion per document hyperparameter (α), and compare each for the case in which all the text from the ad is used for training versus using only entities extracted by the taggers. Hence, we compare six LDA models in a layout denoted as {VEM with estimated alpha, VEM with fixed alpha, Gibbs with estimated alpha} $\times$ {Whole text, Text chunks }.

For the case in which α is estimated during training, we set its initial value to $\alpha = 5/K$ and fix $\beta = 0.1$, as suggested by Griffiths and Steyvers (2004). Then, K is grid-search tuned to minimize perplexity of the model. For the case in which α is fixed, it is grid-search tuned after an optimum K is found. This strategy follows the conclusion that the VEM inference algorithm estimates too low α hyperparameters, as reported by Asuncion et al. (2009). Low α hyperparameters cause the model to assign few topics per document, only one in the worse case.

Dimensionality Selection Each time we fit a mixed-membership model to data, we must specify the number of latent categories, K, in the model. The goal of model selection is to find K^*, the number of latent categories that is optimal in some sense. We use 10-fold cross-validation following the approach described in Airoldi et al. (2010), and widely used in other machine learning applications. First, we split the N job ads into 10 batches. Then, we estimate the model parameters using the ads in nine batches, and we calculate the posterior perplexity of the ads in the tenth held-out batch. This approach leads to summarize how good a model fits for a given $K \in [5, 200]$, on a batch of ads not included in the estimation. We fit each model a total of 60 times (10 times in cross-validation for each of 6 models) for each value of K. Fold splitting during cross-validation was seeded to assure consistency of multiple runs of a model and to assure comparability among different models that use the same data.

For our experiments, we use the LDA R library *topicmodels* by Grün and Hornik (2011), which wraps Blei et al. (2003) C code for VEM inference and Phan et al. (2008) C++ code for Gibbs sampling.

4.5. Topic Coherence

In our coherence experiments, we use the framework proposed by Röder et al. (2015), available online,[2] in which many more scores are available and a reference corpus for probability counts can be specified. Although Mimno et al. (2011) do not use any external reference corpus, Röder et al. (2015) showed that using Wikipedia as an additional reference corpus improved correlation with gold human ratings for this metric. Following this setup, we use as external reference corpus the concatenation of the entire Job Ads dataset (more than 500,000 documents) and the Wikipedia dump in Spanish. Following the literature (Chang et al., 2009; Mimno et al., 2011; Aletras and Stevenson, 2013; Lau et al., 2014), we employ the top 10 words by topic.

5. Results and Discussion

5.1. Skills and Tasks Extraction

Table 3 shows results for the tagger. It can be observed that CARR tagger shows the best performance. This can be explained by the fact that majors are mostly mentioned in determined word patterns in job ads. For the FUN tagger, taking advantage of the fact that functions are not mentioned in the beginning nor the end of the ad improves the precision significantly in comparison to early experiments. In addition, FUN entities mostly appear at the beginning of the sentences.

Entity	# Feat.	P	R	F_1	ACC.
FUN	503701	61.1	62.3	61.7	93.4
REQ	605864	77.6	55.9	65.0	97.1
CARR	215143	87.2	86.9	87.0	99.5

Table 3: Feature set sizes and taggers' performance

5.2. Topic Models Tuning

Following the procedure described in sections 4, we show in Figure 1 the behavior of the held-out perplexity as the number of topics changes. We observe that in general

[2] https://github.com/AKSW/Palmetto

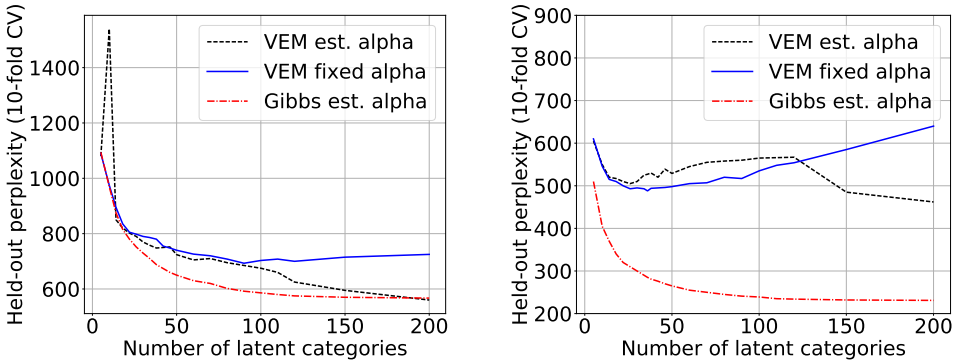


Figure 1: Average held-out perplexity as a function of the number of latent categories K for whole text models 1, 2 and 3 (left), and text chunks models 4, 5 and 6 (right).

there is no agreement among the methods of inference for the optimal number of topics and that in some cases the perplexity does not converge.

Using the *UMass* topic coherence score to measure the quality of the models as the number of topics changes, we observe in Figure 2 that for each method of inference, the optimal number of topics is found between 5 and 18. We choose $K = 10$ as the optimal value for both models, as it gives the best score for models using text chunks (Figure 2, right) regardless of the inference strategy followed. For models using the whole text (left), this value is fairly close to the optimum (15).

5.3. Topic Coherence Improvement

For the optimal number of topics chosen in Section 5.2, 10, the bar plot in Figure 3 shows the improvement of the UMass topic coherence when restricting the text to the chunks extracted by the entity extractors. Also, it can be observed that this happens independently of the method of inference, and that there is at least an improvement of 40% in each case, with *VEM estimated alpha* having the better coherence score when text chunks are used.

5.4. Qualitative and quantitative analysis of inferred categories

Topics are explored by examining the top 10 words (Tables 4, 5 and 6). In addition, the topic proportion for each professional major is investigated. For each major, the mean of posterior membership scores of all documents where this major was required is taken, as proposed by Erosheva et al. (2004). Figure 5 shows this calculation for VEM inference method with fixed alpha. Figure 4 presents matrices for the six

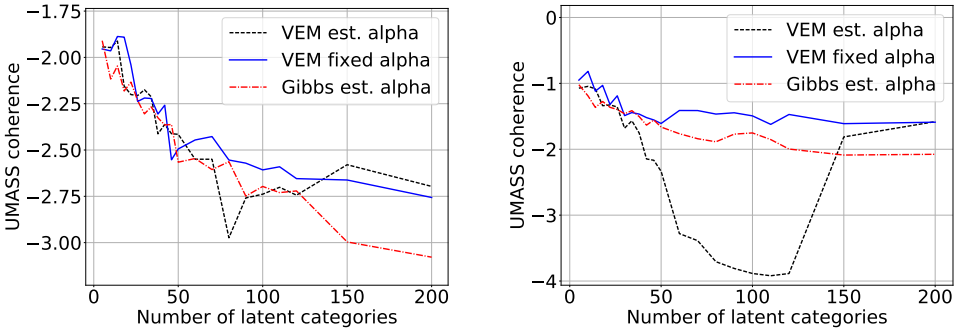


Figure 2: Average *UMass* coherence score (higher is better) as a function of the number of topics K for whole text models 1, 2 and 3 (left), and text chunks models 4, 5 and 6 (right).

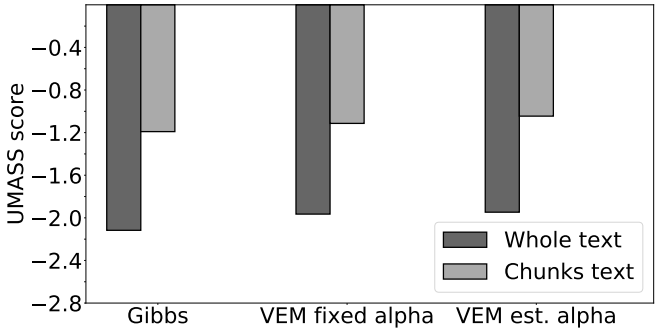


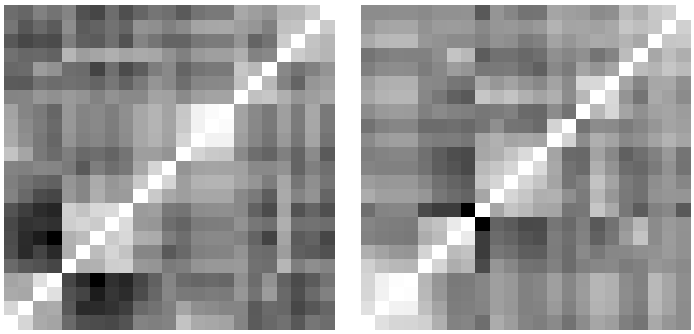
Figure 3: Comparison of the UMass coherence score for each method of inference.

mixed-membership models, which represent the similarity of the probability distributions over categories between all majors. This similarity is calculated using Hellinger distance. Each row and column of each matrix represent a professional major and its similarity with other majors, regarding the text source and inference strategy applied. Major names are not shown because each matrix has different major names order in rows and columns. The purpose of Figure 4 is to unveil the effect of how professional majors are grouped. A similar behavior can be observed in Figure 5 by observing for each topic the majors that have the most vivid colors.

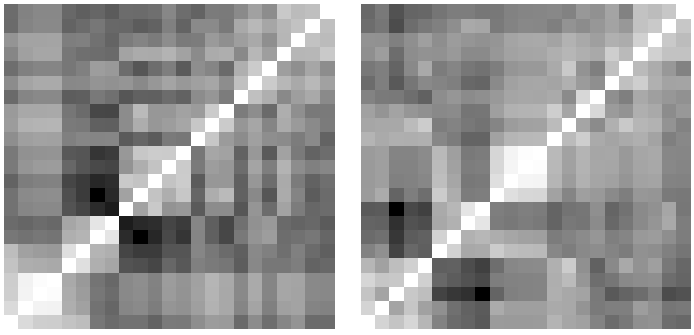
Furthermore, it can be observed in both graphics Figure 5 and 4 that for the case of the text chunks model, getting rid of irrelevant words (ignored by the entity extractors) has the effect of smoothing the probability distribution over topics. For instance, for the whole text model, the job ads for environmental engineering basically just talk about one topic. On the other hand, for the text chunks model, the major now talks about more than one topic with similar proportions.

A closer look at Figure 5 allows to spot three main behaviors under the effect of restricting the source text (whole text versus text chunks).

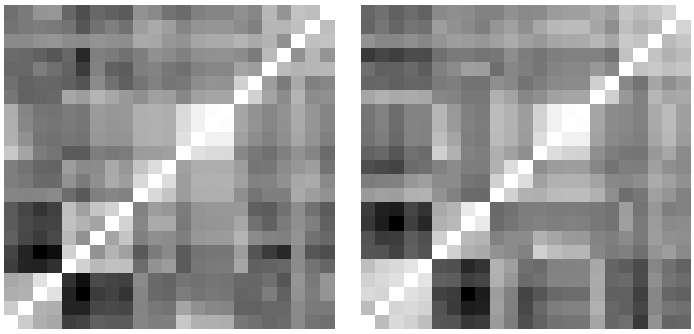
- Joining of redundant categories
  Consider the major of Electronic Engineering. In Figure 5 for the whole text model, topics 4 and 7 are the predominant ones. See Table 4 for the content of the topics. On the other hand, for the text chunks model, it can be seen that only topic 5 is predominant. Table 4 confirms that topic 5 of the text chunks model contains words (with high probability) from both of the topics of the whole text model.
- Splitting in two or more detailed categories
  Consider the majors of Environmental Engineering and Industrial Hygiene and Safety. In Figure 5 for the whole text model, topic 2 is predominant for both majors. Exploration of this topic reveals that its content is related to industrial, environmental safety and management, as can be appreciated in Table 5. On the other hand, for the text chunks model, it can be observed that categories 2 and 10 are predominant and with almost the same proportion. A closer exploration reveals that topic 2 is related to environmental safety and management but no longer contains the word industrial, which appears in topic 10, i.e. the top two words from topic 2 (whole text model) was split.
- Persistence of latent structure
  There are cases where the number of predominant topics does not change. Consider the majors of Systems and Informatics Engineering. For the whole text model, it can be observed that topic 4 is predominant. Likewise, for the text chunks model, topic 9 present the same behaviour. Table 6 shows that the content of these topics is maintained in both models.



(a) VEM inference with estimated alpha, for whole text (left) and text chunks (right) models.

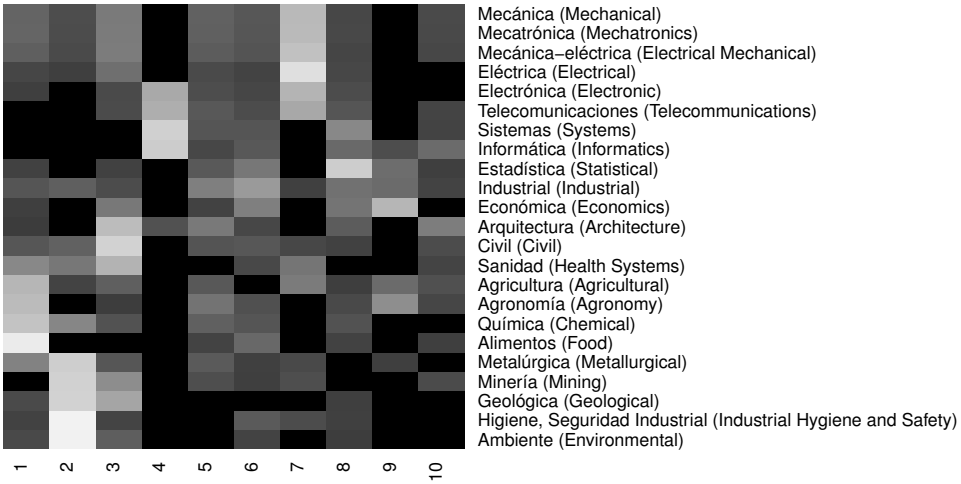


(b) VEM inference with fixed alpha, for whole text (left) and text chunks (right) models.

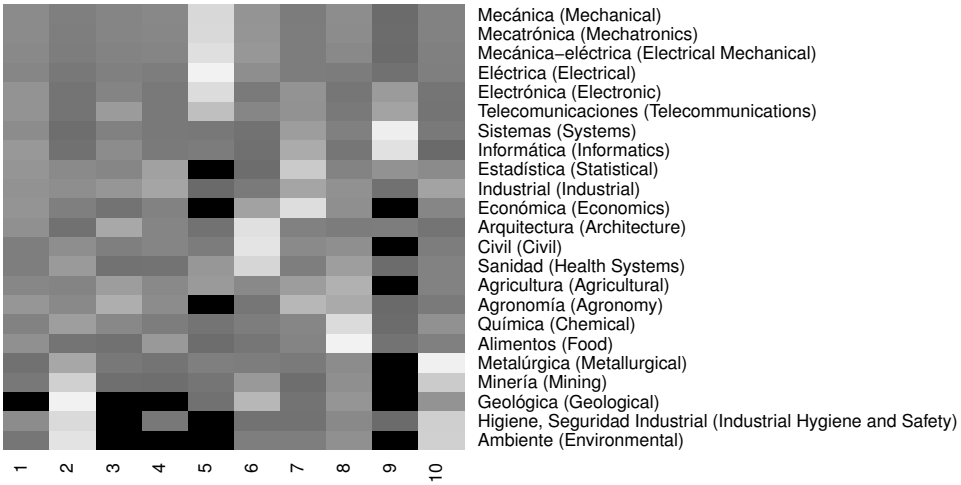


(c) Gibbs inference, for whole text (left) and text chunks (right) models.

Figure 4: Similarity matrices using Hellinger distance between discrete distributions (topic proportion over majors), for each of the six topic models mentioned in section 4.4. A whiter cell means a shorter distance, i.e. more similar categories.



(a) Whole text



(b) Text chunks

Figure 5: Scaled estimated average membership of engineering majors to 10 categories inferred by VEM with fixed alpha for (a) whole text setup and (b) text chunks setup. The whiter the highest the membership; black denotes zero membership. Original Spanish names for majors are showed with the English gloss in parenthesis.

Whole text		Text chunks
Topic 4	Topic 7	Topic 5
sistemas (*systems*)	técnico (*technician*)	mantenimiento (*maintenance*)
técnico (*technician*)	mantenimiento (*maintenance*)	mecánica (*mechanical*)
informática (*informatics*)	mecánica (*mechanical*)	electrónica (*electronics*)
desarrollo (*development*)	eléctrica (*electrical*)	eléctrica (*electrical*)
computación (*computation*)	electricidad (*electricity*)	electricidad (*electricity*)
sql (*SQL*)	industrial (*industrial*)	técnico (*technician*)
programador (*programmer*)	preventivo (*preventive*)	instalación (*installation*)
analista (*analyst*)	electrónica (*electronics*)	reparar (*repair*)
programación (*programming*)	sistemas (*systems*)	preventivo (*preventive*)
servidor (*server*)	instalación (*installation*)	sistemas (*systems*)

Table 4: Topics behavior for VEM fixed α strategy: joining of redundant categories. Each entry consists of the Spanish token and its English gloss in parenthesis.

Whole text		Text chunks
Topic 2	Topic 2	Topic 10
seguridad (*safety*)	seguridad (*safety*)	industrial (*industrial*)
industrial (*industrial*)	risk	supervisor (*supervisor*)
management	environmental	administración (*management*)
ocupacional (*occupational*)	management	marketing
ambiente (*environment*)	ocupacional (*occupational*)	especialización (*specialization*)
supervisor (*supervisor*)	normas (*norms*)	venta (*selling*)
normas (*norms*)	documentos (*documents*)	economía (*economy*)
capacitación (*capacitation*)	seguimiento (*tracing*)	proactivo (*proactive*)
risk	industrial (*industrial*)	responsable (*responsible*)
iso (*ISO*)	soporte (*support*)	dinámico (*dynamic*)

Table 5: Topics behavior for VEM fixed α strategy: splitting in two or more detailed categories. Each entry consists of the Spanish token and its English gloss in parenthesis when applicable.

Whole text	Text chunks
Topic 4	Topic 9
sistemas (*systems*)	sistemas (*systems*)
técnico (*technician*)	informática (*informatics*)
informática (*informatics*)	analista (*analyst*)
desarrollo (*development*)	programador (*programmer*)
computación (*computation*)	sql (*SQL*)
sql (*SQL*)	desarrollo (*development*)
programador (*programmer*)	computación (*computation*)
analista (*analyst*)	programación (*programming*)
programación (*programming*)	servidor (*server*)
servidor (*server*)	administrador (*administrator*)

Table 6: Topics behavior for VEM fixed α strategy: persistence of latent structure. Each entry consists of the Spanish token and its English gloss in parenthesis.

6. Conclusions

Throughout the analysis of multiple variants of topic models, consistent results confirm our hypothesis that coherence of inferred categories significantly improves when using only relevant text extracted by named entity extraction rather that the whole document. In our case study, the relevant text constitutes expected skills, tasks to perform, and academic background in job ads.

Compared to categories inferred using whole-text models, entities models generate categories that join redundant ones and split to high skill-specific categories. In addition, fine-grained categories are preserved with entity models.

Bibliography

Airoldi, E. M., E. A. Erosheva, S. E. Fienberg, C. Joutard, T. Love, and S. Shringarpure. Reconceptualizing the classification of PNAS articles. In *Proceedings of the National Academy of Sciences of the USA*, volume 107, pages 20899–20904, 2010.

Aletras, Nikolaos and Mark Stevenson. Evaluating Topic Coherence Using Distributional Semantics. In *10th Int. Conf. on Computational Semantics (IWCS'13)*, 2013.

Asuncion, A., M. Welling, P. Smyth, and Y. W. Teh. On smoothing and inference for topic models. In *Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence (UAI'09)*, pages 27–34, 2009.

Bikel, D. M., R. Schwartz, and R. M. Weischedel. An Algorithm that Learns What's in a Name. *Journal of Machine Learning*, 34:211–231, 1999.

Blei, D. M. and J. D. Lafferty. Correlated Topic Models. In Weiss, Y., B. Schölkopf, and J. Platt, editors, *Advances in Neural Information Processing Systems*, pages 147–154. MIT Press, Cambridge, 2005.

Blei, D. M. and J. D. Lafferty. Dynamic topic models. In *Proceedings of the 23rd International Conference of Machine Learning (ICML '06)*, pages 113–120, August 2006.

Blei, D. M., A. Ng, and M. I. Jordan. Latent Dirichlet Allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

Blei, D. M., T. L. Griffiths, and M. I. Jordan. The nested Chinese restaurant process and Bayesian nonparametric inference of topic hierarchies. *Journal of the ACM*, 57:7.1–7.30, 2007.

Cardenas Acosta, Ronald, Kevin Bello Medina, Alberto Coronado, and Elizabeth Villota. Engineering job ads corpus, 2016. URL `http://hdl.handle.net/11234/1-2673`. LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Carreras, X., L. Marquez, and L. Padró. Wide-Coverage Spanish Named Entity Extraction. In *VIII Conferencia Iberoamericana de Inteligencia Artificial, IBERAMIA'02*, 2002.

Chang, Jonathan, Sean Gerrish, Chong Wang, Jordan L. Boyd-graber, and David M. Blei. Reading Tea Leaves: How Humans Interpret Topic Models. In Bengio, Y., D. Schuurmans, J. D. Lafferty, C. K. I. Williams, and A. Culotta, editors, *Advances in Neural Information Processing Systems 22*, pages 288–296. Curran Associates, Inc., 2009.

Collins, M. Discriminative training methods for hidden Markov models: Theory and experiments with perceptron algorithm. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2002.

Erosheva, E. A., S. E. Fienberg, and J. Lafferty. Mixed-membership models of scientific publications. In *Proceedings of the National Academy of Sciences of the USA*, volume 101, pages 5220–5227, 2004.

Griffiths, T. L. and M. Steyvers. Finding scientific topics. In *Proceedings of the National Academy of Sciences of the USA*, volume 101, pages 5228–5235, 2004.

Grün, Bettina and Kurt Hornik. topicmodels: An R Package for Fitting Topic Models. *Journal of Statistical Software*, 40(13):1–30, 2011. doi: 10.18637/jss.v040.i13.

Hall, Mark M., Paul D. Clough, and Mark Stevenson. Evaluating the Use of Clustering for Automatically Organising Digital Library Collections. In *Second International Conference on Theory and Practice of Digital Libraries (ERCIMDL)*, 2012.

Lafferty, John D., Andrew McCallum, and Fernando C. N. Pereira. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In *Proceedings of ICML*, pages 282–289, San Francisco, CA, USA, 2001. ISBN 1-55860-778-1.

Lau, Jey Han, David Newman, and Timothy Baldwin. Machine Reading Tea Leaves: Automatically Evaluating Topic Coherence and Topic Model Quality. In *European Chapter of the Association for Computational Linguistics (EACL'14)*, 2014.

Liang, P. and M. Collins. Semi-supervised learning for natural language. Master's thesis, Massachusetts Institute of Technology, 2005.

Miller, S., J. Guinness, and A. Zamanian. Name tagging with word clusters and discriminative training. In *Proceedings of the Proceedings of HLT-NAACL 2004*, pages 337–342, 2004.

Mimno, David M., Hanna M. Wallach, Edmund M. Talley, Miriam Leenders, and Andrew McCallum. Optimizing Semantic Coherence in Topic Models. In *Empirical Methods in Natural Language Processing (EMNLP 2011)*, 2011.

Musat, Claudiu Cristian, Julien Velcin, Stefan Trausan-Matu, and Marian-Andrei Rizoiu. Improving Topic Evaluation Using Conceptual Knowledge. In *22nd International Joint Conference on Artificial Intelligence (IJCAI-2011)*, 2011.

Newman, D., C. Chemudugunta, and P. Smyth. Statistical entity-topic models. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 680–686, August 2006.

Newman, David, Jey Han Lau, Karl Grieser, and Timothy Baldwin. Automatic Evaluation of Topic Coherence. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, HLT '10, pages 100–108, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics. ISBN 1-932432-65-5. URL `http://dl.acm.org/citation.cfm?id=1857999.1858011`.

Nguyen, Thang, Jordan L. Boyd-Graber, Jeffrey Lund, Kevin D. Seppi, and Eric K. Ringger. Is Your Anchor Going Up or Down? Fast and Accurate Supervised Topic Models. In *North American Chapter of the Association for Computational Linguistics (NAACL 2015)*, 2015.

Paul, Michael J. and Roxana Girju. A Two-Dimensional Topic-Aspect Model for Discovering Multi-Faceted Topics. In *24th Annual Conference on Artificial Intelligence (AAAI-10)*, 2010.

Phan, Xuan Hieu, Minh Le Nguyen, and Susumu Horiguchi. Learning to classify short and sparse text & web with hidden topics from large-scale data collections. In *17th International World Wide Web Conference (WWW 2008)*, pages 91–100, 2008.

Ramshaw, L. A. and M. P. Marcus. Text chunking using transformation-based learning. In *Proceedings of the Third Workshop on Very Large Corpora*. ACL, 1995.

Reisinger, Joseph, Austin Waters, Bryan Silverthorn, and Raymond J. Mooney. Spherical Topic Models. In *27th International Conference on Machine Learning (ICML 2010)*, 2010.

Röder, Michael, Andreas Both, and Alexander Hinneburg. Exploring the Space of Topic Coherence Measures. In *Proceedings of WSDM*, 2015.

Stevens, Keith, W. Philip Kegelmeyer, David Andrzejewski, and David Buttler. Exploring Topic Coherence over Many Models and Many Topics. In *Proceedings of EMNLP-CoNLL'12*, 2012.

Teh, Y. W., M. I. Jordan, M. J. Beal, and D. M. Blei. Hierarchical Dirichlet processes. *Journal of the American Statistical Association*, 101:1566–1581, 2006.

Yang, Yi, Doug Downey, and Jordan L. Boyd-Graber. Efficient Methods for Incorporating Knowledge into Topic Models. In *Empirical Methods in Natural Language Processing*, 2015.

Address for correspondence:
Ronald Cardenas
`racardenasa@uni.pe`
National University of Engineering,
Department of Mechanical Engineering
Tupac Amaru Avenue 210, Lima 25, Lima, Peru

The Prague Bulletin of Mathematical Linguistics
NUMBER 110 APRIL 2018

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`.