PBML

The Prague Bulletin of Mathematical Linguistics

NUMBER 105 APRIL 2016

EDITORIAL BOARD

Published twice a year by Charles University in Prague

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 105 APRIL 2016

CONTENTS

PBML

The Prague Bulletin of Mathematical Linguistics

NUMBER 105 APRIL 2016

EDITORIAL

Life Anniversary of Petr Sgall, the Founder of PBML

The Editors of The Prague Bulletin of Mathematical Linguistics wholeheartedly join the co-workers, former students and broader linguistic community to celebrate this year's life anniversary of its founding Editor-in-Chief Professor PhDr. Petr Sgall, DrSc. Dr.h.c mult. (born May 27th, 1926). Petr Sgall is an outstanding member of the Czech linguistic community highly appreciated at home and abroad. His scientific interests are extremely broad: they range from Indoeuropean studies through topical issues of Czech grammar and language culture to theoretical and computational linguistics. He is the author of the original functionally oriented framework of formal description of grammar, called Functional Generative Description, which stands as an alternative to the Chomskyan concept of generative grammar. He is one of the founders of Czech(oslovak) computational linguistics, the high level of which he succeeded to retain even under the unfavourable conditions of the restrictive political regime of the past. He received international recognition as an elected member of the Academia Europaea and was elected a honorary member of the Linguistic Society of America. He has got two honorary doctorates, one by the Hamburg University and one by the French INALCO institute in Paris.

To recall briefly some of his chief research interests, we reprint here the Introduction to a volume of Petr Sgall's selected writings called Language in Its Multifarious Aspects published in 2006 by Karolinum Publishing House in Prague.

INTRODUCTION TO SELECTED PAPERS OF PETR SGALL

Language in its multifarious aspects (Prague, Karolinum, 2006)

Eva Hajičová and Jarmila Panevová

Petr Sgall (born May 27th, 1926 in České Budějovice, but spending most of his childhood in the small town Ústí nad Orlicí in eastern Bohemia and living since his university studies in Prague) is one of the most prominent Czech linguists belonging to the so-called "second generation" of the world-famous structural and functional Prague School of Linguistics. His first research interests focused on typology of languages, in which he was a pupil of Vladimír Skalička. His PhD thesis was on the development of inflection in Indo-European languages (published in Czech in 1958b). He spent a year of postgraduate studies in Cracow, studying with J. Kuryłowicz. He habilitated as docent (associate professor) of general and Indoeuropean linguistics at Charles University in 1958 on the basis of his Cracow study of infinitive in Old Indian (Infinitive im R̥gveda, published the same year).

Since his beginnings, he was always deeply interested in the exceptional situation of Czech where alongside with the standard form of language there exists a form of Czech that is usually called ‚Common Czech' (as it is not restricted to some geographical area as dialects are) and that is used by most Czech speakers in everyday communication. In this he was influenced by the work of Bohuslav Havránek on functional stratification of Czech.

At the beginning of the 1960s, Sgall was one of the first European scholars who got acquainted with the emerging new linguistic paradigm, Chomskyan generative grammar. On the one hand, he immediately understood the importance of an explicit description of language, but at the same time, he was aware that the generative approach as presented in the early days of transformational grammar, lacks a due regard to the functions of language (at this point we want to recall his perspicacious analysis of Prague School functionalism in his paper published in 1964 in the renewed series Prague Linguistic Circle Papers (pre-war TLCP), the Travaux linguistiques de Prague Vol. I in 1964. Based on the Praguian tenets, Sgall formulated and developed an original framework of generative description of language, the so-called Functional Generative Description (FGD). His papers in the early sixties and his book presenting FGD (Sgall, 1967) were the foundation stones of an original school of theoretical and computational linguistics that has been alive and flourishing in Prague since then. Sgall's innovative approach builds on three main pillars: (i) dependency syntax, (ii) information structure as an integral part of the underlying linguistic structure, and (iii) due regard to the distinction between linguistic meaning and cognitive content.

Petr Sgall has proved also outstanding organizational skills. In 1959, he founded a small subdepartment of mathematical linguistics (called then ‚algebraic', to get distinguished from the traditional quantitative linguistics) and theory of machine translation at the Faculty of Arts of Charles University, followed by a foundation of a small group of computational linguistics also at the Faculty of Mathematics and Physics (in 1960) of the same University. In 1968, the two groups were integrated under his leadership into the Laboratory of Algebraic Linguistics, attached to the Faculty of Arts. This Laboratory, due to the political changes in the country caused by Russia-led invasion, had, unfortunately, a very short life-span. In 1972, Sgall faced a forced dismission from the University for political reasons, and the whole group was eventually doomed to be dissolved. Fortunately, thanks to a group of brave colleagues and friends at the Faculty of Mathematics and Physics, he and his collaborators were transferred to this Faculty, less closely watched (by guardians of ideology) than was the domain of the Humanities. Even there, however, the conditions were not at all easy for him – for several years, the Communist Party decision for the group to disappear was in power, the number of Sgall's collaborators was harshly reduced and many obstacles were laid in the way of research in computational linguistics as such. Sgall himself was deprived of possibilities to teach, supervise students, travel to the West, attend conferences there, and only slowly and gradually he could resume some of his activities in the 1980s. Nevertheless, not only the core of the research group continued working in contact with Western centres and their leading personalities (as evidenced above all by the contributions to his Festschrift edited by Jacob Mey and published by John Benjamins in 1986), but it was also possible to help three other immediately endangered colleagues to survive at the University.

The years after the political changes in our country in 1989 have brought him a due satisfaction after the previous years of suppression: a possibility of a 5-month stay as a research fellow at the Netherlands Institute of Advanced Studies in Wassenaar (a standing invitation he has had for many years but which he was not allowed to accept for political reasons), the membership in the prestigious Academia Europaea, the International Research Prize of Alexander von Humboldt in 1992, a visiting professorship at the University in Vienna in 1993, the Prize of the Czech Minister of Education in the same year, a honorary doctorate at the Institut National des Langues et Civilisations Orientales in Paris in 1995 and at the Hamburg University in 1998 and an honorary membership in the Linguistic Society of America in 2002, not to speak about numbers of invitations for lectures and conferences in the whole world, from the U.S.A. to Malaysia and Japan. As a Professor Emeritus of Charles University since 1995, he is still actively involved in teaching and supervising PhD students, in participating at Czech and international research projects and in chairing the Scientific Board of the Vilém Mathesius Center he helped to found in 1992.

Petr Sgall was also among those who helped to revive the Prague Linguistic Circle already in 1988 and has a substantial share in reviving also the book series Travaux de Cercle linguistique de Prague (under a parallel title Prague Linguistic Circle Papers), the first volume of which appeared in 1995 (published in Amsterdam by John Benjamins Publ. Company) and the fifth volume is now in preparation.

With his research activities based on a true Praguian functional approach, he thus more than made up for his negative attitudes published in the beginning of the fifties, a revolutionary and rash approach to which he was inspired by his wartime experience (his father died in Auschwitz, as did eleven of his closest relatives, and Petr Sgall himself spent some months in a labour camp) and ill-advised by some of his tutors. Let us remind in this connection e.g. his review of three American volumes devoted to the Prague School published in 1978 in the Prague Bulletin of Mathematical Linguistics (a University periodical founded by Sgall in 1964), at the time when the political situation in the country and his own personal position was very difficult.

The present volume is conceived of as a reflection of the broad scope of Petr Sgall's linguistic interests, and, at the same time, as a document how lively the Prague School tenets are if developed by such a creative personality. Also, the contributions included in the volume illustrate characteristic features of Petr Sgall as a researcher: the overwhelming variety of deeply rooted topics of interest, the ability to penetrate into the substance of arguments and giving a convincing counterargument, the consistence of opinions but, at the same time, open-mindedness and openness to discussion and willingness to accept the opponent's viewpoint if he finds good reasons for it. There are not many researchers of his position who would be able to react so creatively to stimuli from the outside, to learn a lesson from them and to push his students to do the same ('read if you want to be read' is one of his favourite slogans).

Sgall's papers selected for this volume have been sorted in six parts covering both general theoretical questions of language typology, linguistic description, relationships of grammar, meaning and discourse as well as more specific topics of the sentence structure and semantics. It is a matter of course that we could not omit at least a small sample of contributions to his most beloved child, functional stratification of Czech and orthography. Below, we give a very brief outline of the main views as present in the papers; we refer to the individual papers by their serial numbers in brackets.

Part A (**General and Theoretical Issues**) provides a broader picture of Sgall's understanding of the tenets of Prague School Linguistics and their reflection in the present-day development of language theories, including a brief characterization of the Functional Generative Description, based on a perspicuous account of the topic-focus articulation and on dependency syntax [4]. Sgall has always been aware of the usefulness of comparison of linguistic frameworks and approaches [3]. His original formal approach called Functional Generative Description (FGD) was presented in a comparative perspective in the context of M. A. K. Halliday's Systemic (Functional) Grammar [5]. FGD was proposed as early as in the mid-sixties [9] and was conceived of as an alternative to Chomskian generative transformational grammar. It is based on the dependency approach to syntax (8; this paper, in spite of its title, presents a proposal how to generate underlying dependency structures and is not concerned only with topic-focus articulation) and on a firm conviction that what constitutes the syntax of the sentence is its underlying structure rather than its surface shape [7]. As a founder of computational linguistics in Prague (and in the whole of former Czechoslovakia), he has always been very sensitive to put a right balance to the formal and empirical aspects of that interdisciplinary domain [6]. In this connection it should be recalled that Petr Sgall used his involuntary shift from the Faculty of Arts to the Faculty of Mathematics and Physics in the years after the Russian invasion in a fruitful way: not only he has won the interest of several young computer scientists in computational and theoretical linguistics, thus helping to establish this field as one of the curriculum specialities at this Faculty, but also offered a "shelter" and research environment to those whose political background was not "reliable" enough to apply for admission at an ideologically oriented Faculty of Philosophy but whose skills enabled them to be admitted to a less "watched" Faculty of Mathematics and Physics. It is symptomatic for the atmosphere of that time and for Sgall's sharp eyes and good intuitions that most of these former students belong now to promising researchers and university teachers at both of the Faculties.

The other fundamental issue Sgall has been recently concentrating on is the relation of the core of language and its periphery [1], [2]. These notions are also rooted in the Prague School tradition, but Sgall puts them into a broader and more complex perspective. He claims that since language is more stable in its core, regularities in language should be searched for first in this core; only then it is possible to penetrate into the subtleties and irregularities of the periphery. The relatively simple pattern of the core of language (in Sgall's view, not far from the transparent pattern of propositional calculus) makes it possible for children to learn the regularities of their mother tongue. The freedom of language offers space for the flexibility of the periphery.

Petr Sgall gives an impression of a most serious, matter-of-fact and sober person. To document that he understands good and intelligent humour and that he is creative also in this respect, we include in the present volume his "Mourphology" paper [10] as a kind of delicatesse.

Parts B and C focus on two fundamental pillars of Sgall's linguistic theory: underlying dependency syntax (Part B) and information structure (topic-focus articulation) as a basic aspect of the sentence (Part C).

Section B (**Syntax**) contains papers extending and examining the main issues of the Functional Generative Description (FGD), proposed by the author in the 1960s, [11], [12], [13]. The papers chosen for this section present the author's argumentation for the importance of the difference between linguistic meaning and ontological content, which delimits the opposition of language as a system and the domain of cognition. P. Sgall demonstrates in [13] that this distinction, known since F. de Saussure and L. Hjelmslev (with linguistic meaning characterized as "form of content"), can be determined with the help of operational and testable criteria. On such a basis, the "deep cases" (case roles, i.e. the underlying, tectogrammatical syntactic relations) can be specified as belonging to the language patterning and differentiated from a conceptualization of the scenes more clearly than with many other approaches, including that of Ch. Fillmore. Strict synonymy is understood as a condition of tectogrammatical identity. Open questions (more or less directly connected with empirical studies of texts and corpora), remaining in the specification of the list of arguments (participants) and adjuncts, are discussed in [12], where also relations other than dependency are investigated. Sgall points out the possibility to linearise even rather complex more-dimensional graphs representing projective tectogrammatical structures (including coordination and apposition) into relatively simple strings of complex symbols with a single kind of parentheses. He claims that this type of structure comes close to elementary logic and thus documents that the core of language exhibits a pattern based on general human mental capacities, which might be useful in analysing the acquisition of the mother tongue by children. The author's subtle sense for the development of linguistic research is reflected by his participation in conceiving and constructing the Prague Dependency Treebank, a syntactically annotated part of the Czech National Corpus. P. Sgall describes the main issues of the procedure of the syntactic annotation based on FGD in [11]. Examples of tectogrammatical tree structures are given here and an outlook for the future extension of the automatic part of the procedure is discussed.

One of the most innovative contributions of Petr Sgall to theoretical and formal linguistics is his claim that the **topic-focus articulation** (TFA, Part C, see also [4]) of the sentence is semantically relevant and constitutes the basic sentence structure essential for the semantic interpretation of the sentence. As discussed now in Hajičová and Sgall (in prep.) more explicitly than before, this dichotomy is considered to be more fundamental than the subject–predicate structure of traditional grammar and of the "mainstream" theories (be it analysed in terms of constituents or of dependency syntax). Sgall refers back to Aristotelian original understanding of 'subject' as 'given by the circumstances' (τὸ ὑποκε μενον – translated in Gemoll's 1908 dictionary as *die gegebenen Verhältnisse* 'the given circumstances' and 'predicate' (τὸ κατηγορο μενον – *das Ausgesagte* 'the enounced') as what is 'predicated' about the 'subject', emphasizing the aboutness relation. It is in this sense that the content of an utterance (i.e. of a sentence occurrence) can be properly seen in the interactive perspective, as an operation on the hearer's memory state. It should be noticed that the first paper by Sgall on TFA and its inclusion into a generative description of language was published as early as in 1967 [17]. The surface word order is conceived of in relation to TFA; the differences between the surface and underlying order of items of the sentence can be accounted for by a relatively small number of 'movement' rules. The study of issues related to the information structure of the sentence is paid a serious attention in the Prague School history introduced there by the studies of Vilém Mathesius in the first half of last century and continued by Jan Firbas, whose approach is critically examined from the FGD viewpoint in [14]. A study of these issues was given a more intensive attention by a wider linguistic community only later in the last two decades of 20th century and it is thanks to Sgall that the position of the Czech studies on the international scene has been duly specified [15] and, even more importantly, that the attention has been focussed on the basic semantic relevance of these issues [14].

Part D (**From sentence to discourse in semantics**) gives a perspective on Sgall's views on the delimitation of the language system (linguistic competence) against the domain of cognition and the process of communication. He analyses issues going beyond the limits of the sentence – both in the 'dimensional' sense (extending the scope of attention to discourse) and in the sense of crossing the boundaries of the literal meaning towards the issues of reference, cognitive content and truth conditions. Well aware of the distinction between linguistic meaning and (extra-linguistic) content claimed by Praguian scholars following de Saussure, Sgall [19] analyses the notion of 'meaning' as present in linguistic and logical discussions and suggests to distinguish between several explicata of the concept: (a) meaning as linguistic patterning (literal meaning), (b) meaning (or sense) as literal meaning enriched by reference, which can be understood as a layer of interface between linguistic structure and the semantic(-

pragmatic) interpretation of natural language, (c) meaning in the sense of structured meaning, i.e. with specifications more subtle than propositions (Lewis-type meaning), (d) meaning as intension, (e) meaning as extension, and (f) meaning as content, taking into account the context-dependence of the content of the utterance. In this paper, as well as in all other papers on the issues of meaning, especially when discussing the distinction between ambiguity and vagueness, a crucial emphasis is laid on the necessity to establish and apply operational criteria for making the relevant distinctions. Sgall's own proposal of a starting point for a description of the semantic system of a language is presented in [20] as a nine-tuple, taking into account the outer shape of the sentence described, the representation(s) of the meaning(s) of the sentence, the entities that can be referred to, the set of items activated (salient) at the given point of time of the discourse, the possible sense(s) of the utterance token with the given meaning, the class of possible worlds, the set of truth values, and Carnapian proposition (i.e. a partial function from Sense(Meaning(Sentence)) into the class of functions from the possible worlds into the truth values). The author tests the potential of the proposed framework on several examples, each illustrating some particular point present in the discussions of natural language semantics such as the relevance of topic-focus articulation (see [4] and Part C of the volume) for semantic interpretation, the importance of the different kinds of contexts (attitudinal, quotational) for the operational criteria for synonymy, and the cases of presupposition failure and contradictions. Discourse patterning in its dynamic perspective based on the notion of the hierarchy of activation is discussed in detail in [18] and partly also already in [20].

The papers included in part E (**Typology of languages**) are closely connected with the author's linguistic beginnings. As a pupil of V. Skalička, the founder of the Prague School typology, Sgall develops the ideas of his teacher and supervisor in [22] and [23] (see also [1]), pointing out that each of the types of languages can be understood as based on one fundamental property, which concerns the way of expression of grammatical values: by free or affixed morphemes, by a word-final alternation (a single ending), or by word order. In [24], which is a part of Sgall's habilitation about the infinitives in the Ṛgveda, the nominal and verbal characteristics of infinitive in agglutinative and inflectional languages are analysed. While in languages of the former type the role of the "second verb" in a sentence is fulfilled first of all by verbal nouns, the latter type prefers an infinitive with a single ending (without preposition), and the analytical counterpart is a subordinate clause. In [23] the author discusses various meanings in which the terms "type" and "typology" are used in contemporary linguistics, distinguishing between polysemy of a term and different views of a single object of analysis. A type differs from a class in that it is based on a cluster of properties, on their "extreme combination". Working with one fundamental property for each type and with the probabilistic implication makes it superfluous to enumerate sets of properties defining the individual types. Agglutinative and inflectional languages are compared as for their "naturalness" (Natürlichkeit) in [21]. Although inflection, based on a single ending with many irregularities, seems less natural than agglutination from the morphemic point of view, inflection conveys a more appropriate basis for natural syntax (with cases rendering mainly arguments or theta roles, the high degree of "free" word order expressing the topic-focus articulation, and analytical prepositions occurring in the forms of adverbials). Sgall, as always, is aware that some questions examined here are far from a finite solution (e.g. the boundaries between lexical units and syntagms or between word derivation and morphemics are still open for further discussion).

The papers included in Part F (**Speaking and writing**) reflect Sgall's permanent interest in sociolinguistic issues. The situation of Czech in everyday speech is characterized by the author as code switching rather than diglossia known e.g. from the Arabic world. Following the classical functional viewpoint of the Prague Linguistic Circle, Sgall suggests that linguists to describe the actual usage of Czech (especially of its morphemics, considered to be the main source of the differences between the varieties of Czech) in different layers of communication, rather than to impose prescriptions. The position of Common Czech among the varieties differs nowadays from that of the so-called interdialects. Speakers of Czech are encouraged by the author to reduce the means with a bookish flavour in their communication, because their occurrence in other than bookish contexts is one of the reasons why the Standard norm and everyday spoken Czech are quite distant. The nature of the orthographical systems using graphemes is studied in [26], where the author provides a definition of such notions as alphabet, orthography and spelling, based first of all on the relation between phonemes and graphemes. Questions about appropriateness of orthographical systems are formulated on the basis of this explicit description. Sociolinguistic issues connected with an orthographical reform are touched upon by the author as well.

It is not only the broad scope of interests and deep insights that characterize Petr Sgall as an outstanding scientific personality. His deep knowledge and clear view of linguistic (and, in a broader sense, cultural) resources and background ranging from the historical beginnings up to the present-day modern trends is in a unique balance with the originality of his own proposals and solutions. He has never fallen into the trap of black-and-white descriptions of language phenomena: he has always been aware of the restrictions given by the complexity of the described object, i.e. language, and has found a reasonable way out by distinguishing between the notions of the centre (core) of the system and those of the system's periphery. Sgall's deep insights and capability to distinguish these two aspects is documented by his contributions throughout the present volume.

References

The first part of this section contains numbered references to Petr Sgall's writings referred to in the above Introduction and contained in the volume Multifarious Aspects of Language, Karolinum, Prague 2006. The second part contains all other references mentioned in the Introduction.

Part I: Writings of Petr Sgall referred to by numbers in the Introduction

[1] Types of Languages and the Simple Pattern of the Core of Language. In P. Sterkenburg (ed.), *Linguistics Today – Facing a Greater Challenge* (Plenary lectures from the 17th International Congress of Linguists. Amsterdam – Philadelphia: John Benjamins, 243–265.

[2] Freedom of language: Its nature, its sources and its consequences. *Prague Linguistic Circle Papers* 4. Amsterdam – Philadelphia: Benjamins, 2002, 309–29.

[3] On comparison of approaches (Remarks and illustrations). *Linguistica Pragensia* 2000:73–84.

[4] Functionalism in Czech linguistics and in the world. *Linguistica Pragensia* 1997, 64–81.

[5] Structure, meaning and use. In: Anne-Marie Simon-Vandenbergen, Kristin Davidse and Dirk No l (eds.): *Reconnecting language: Morphology and syntax in functional perspectives.* Amsterdam – Philadelphia: John Benjamins, 1997, 73–98.

[6] Formal and computational linguistics in Prague. In: *Prague Linguistic Circle Papers* 1, Amsterdam – Philadelphia: John Benjamins, 1995, 23–35.

[7] Underlying structure of sentences and its relations to semantics. In: T. Reuther (ed.), *Wiener Slawistischer Almanach*. Sonderband 33. Wien: Gesellschaft zur Förderung slawistischer Studien, 1992, 273–282.

[8] A dependency based specification of topic and focus II - Formal account. *SMIL, Journal of Linguistic Calculus*, 1980, No. 1-2, 110–140.

[9] Generative Beschreibung und die Ebenen des Sprachsystems. In: *Zeichen und System der Sprache III, Schriften zur Phonetik, Sprachwissenschaft und Kommunikationsforschung* Nr. 11, Berlin 1966, 225–239.

[10] Introduction to Linguistic Mourphology. PBML 48, 1987, 77-80. Printed in *Journal of Pragmatics* 13, 1989:1015–1018.

[11] Underlying structures in annotating Czech National Corpus. In: G. Zybatow, U. Junghanns, G. Mehlhorn and L. Szucsich (eds.), *Current issues in formal Slavic linguistics*. Frankfurt/M.: Peter Lang, 2001, 499–505.

[12] Revisiting the classification of the dependents. In: E. Hajičová (ed.), *Issues of valency and meaning. Studies in honour of Jarmila Panevová*. Prague: Karolinum, 1998, 15–26.

[13] Case and meaning. *Journal of Pragmatics* 4, 1980, 525–536.

[14] From functional sentence perspective to topic focus-articulation. In: J. Hladký (ed.), *Language and function. To the memory of Jan Firbas*. Amsterdam – Philadelphia: Benjamins, 2003, 279–287.
[15] The position of Czech linguistics in theme-focus research. In: R. Steele and T. Threadgold (eds.). *Language Topics*. Amsterdam – Philadelphia: John Benjamins, 1987, 47–55.
[16] Wortfolge und Fokus im Deutschen. In: W. Abraham und G. Narr, (eds.), *Satzglieder im Deutschen*. Tübingen 1982, 59–74.
[17] Functional sentence perspective in a generative description. *Prague Studies in Mathematical Linguistics*, 2, 1967, 203–225.
[18] Dynamics in the meaning of the sentence and of discourse. In: J. Peregrin (ed.), *Meaning: The dynamic turn*. Oxford: Elsevier Science Ltd., 2003, 169–184.
[19] From meaning via reference to content. In: James Hill and Petr Koťátko (eds.), *Karlovy Vary Studies in Reference and Meaning*. Prague: Filosofia Publications, 1995, 172–183.
[20] Meaning, reference and discourse patterns. In: Ph. Luelsdorff (ed.). *The Prague School of Structural and Functional Linguistics*. Amsterdam – Philadelphia: John Benjamins, 1994, 277–309.
[21] Natürlichkeit, Syntax und Typologie. *Zeitschrift für Phonetik, Sprachwissenschaft und Kommunikationsforschung* 41, 1988, 463–470.
[22] Die Sprachtypologie V. Skaličkas. In: P. Hartmann (ed.), *V. Skalička: Typologische Studien*. Schriften zur Linguistik 11, Braunschweig-Wiesbaden: Vieweg, 1979, 1–20.
[23] On the notion "type of language". *Travaux linguistiques de Prague* 4, 1971, 75–87.
[24] *Zur Typologie des Infinitives*. A chapter from Die Infinitive im Rgveda. AUC Philologica 2-3, 1958, 137-268. Allgemeine Fragen, 137–142,
[25] Spoken Czech revisited. In: L. A. Janda, R. Feldstein and S. Franks (eds.), *Where one's tongue rules well. A festschrift for Ch. Townsend*. Indiana Slavic Studies 13, 2002, 299–309
[26] Towards a theory of phonemic orthography. In: P. Sgall and P. Zima (eds.), *Questions of orthography and transliteration*. Explizite Beschreibung der Sprache und automatische Textbearbeitung 12. Prague: Charles University 1986, 1–46. Reprinted in: Philip A. Luelsdorff (ed.), *Orthography and phonology*. Amsterdam: John Benjamins, 1987, 1–30.

II: References from the Introduction

This list of references contains only papers and books referred to by the authors of the Introduction. Petr Sgall's bibliography before 1986 was compiled as a gift from his colleagues at the occasion of his 60th birthday and was made available as an internal report of the Faculty of Mathematics and Physics, Charles University; the bibliographical data from later periods were published at the occasions of his birthday in the Prague Bulletin of Mathematical Linguistics (PBML) 55, 1991, 95-98; PBML 65-66, 1996, 113-122 (bibliography 1986-1996, with a short introduction "Petr Sgall Septuagenerian") and PBML 75, 2001, 87-91 (bibliography 1996-2000). A complete bibliography of Petr Sgall is attached at the end of the volume Multifarious Aspects of Language, Prague: Karolinum, 2006.

Bibliography

Gemoll, W. *Griechisch-deutsches Schulwörterbuch*. Freytag, Vienna: Tempsky – Leipzig, 1908.

Hajič, Jan. Building a Syntactically Annotated Corpus: The Prague Dependency Treebank. In *Issues of Valency and Meaning. Studies in Honour of Jarmila Panevová (ed. Eva Hajičová)*, pages 106–132. Karolinum, Charles University Press, Prague, ISBN 80-7184-601-5, 1998.

Hajič, Jan, Barbora Vidová Hladká, Alena Böhmová, and Eva Hajičová. The Prague Dependency Treebank: A Three-Level Annotation Scenario. In *Treebanks: Building and Using Syntactically Annotated Corpora (ed. Anne Abeille)*. Kluver Academic Publishers, v tisku, 2000.

Hajič, Jan, Barbora Vidová Hladká, Jarmila Panevová, Eva Hajičová, Petr Sgall, and Petr Pajas. Prague Dependency Treebank 1.0. CDROM, LDC2001T10., ISBN 1-58563-212-0, 2001. Linguistic Data Consortium, Philadelphia. University of Pennsylvania.

Hajicová, Eva. Dependency-based underlying-structure tagging of a very large Czech corpus. *TAL. Traitement automatique des langues*, 41(1):57–78, 2000.

Hajičová, Eva, Petr Sgall, and Barbara Partee. Topic-focus articulation, tripartite structures, and semantic content. Kluwer, Dordrecht, ISBN 0-7923-5289-0, 1998.

Mey, J., editor. *Language and Discourse: Test and Protest. A Festschrift for Petr Sgall*. John Benjamins Publ. Company, Amsterdam-Philadelphia, 1986.

Panevová, Jarmila, Eva Hajičová, and Petr Sgall. K nové úrovni bohemistické práce: Využití anotovaného korpusu. [Towards a new level of work in the study of Czech: Working with an annotated corpus]. *Slovo a slovesnost*, 63:161–177, 241–262, 2002. ISSN 0037-7031.

Sgall, Petr. Die Infinitive im R̥gveda. *AUC-Philologica*, 2–3:137–268, 1958a.

Sgall, Petr. *Vývoj flexe v indoevropských jazycích, zejména v češtině a v angličtině [The development of inflection in Indo-European languages]*. Rozpravy ČSAV, Prague, 1958b.

Sgall, Petr. Zur Frage der Ebenen im Sprachsystem. *Travaux linguistiques de Prague*, 1:95–106, 1964.

Sgall, Petr. *Generativní popis jazyka a česká deklinace [Generative Description of Language and Czech Declension]*. Academia, Prague, 1967.

Sgall, Petr. Three American volumes connected with Czech linguistics. *PBML*, 30:61–68, 1978.

Sgall, Petr, Eva Hajičová, and Jarmila Panevová. *The Meaning of the Sentence in Its Semantic and Pragmatic Aspects*. Reidel, Dordrecht, 1986. (Prague: Academia).

PBML

The Prague Bulletin of Mathematical Linguistics
NUMBER 105 APRIL 2016 17–50

CzEngVallex: a Bilingual Czech-English Valency Lexicon

Zdeňka Urešová, Eva Fučíková, Jana Šindlerová

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

Abstract

This paper introduces a new bilingual Czech-English verbal valency lexicon (called CzEngVallex) representing a relatively large empirical database. It includes 20,835 aligned valency frame pairs (i.e., verb senses which are translations of each other) and their aligned arguments. This new lexicon uses data from the Prague Czech-English Dependency Treebank and also takes advantage of the existing valency lexicons for both languages: the PDT-Vallex for Czech and the EngVallex for English. The CzEngVallex is available for browsing as well as for download in the LINDAT/CLARIN repository.

The CzEngVallex is meant to be used not only by traditional linguists, lexicographers, translators but also by computational linguists both for the purposes of enriching theoretical linguistic accounts of verbal valency from a cross-linguistic perspective and for an innovative use in various NLP tasks.

1. Introduction

The CzEngVallex lexicon[1] is a result of the project called "A comparison of Czech and English verbal valency based on corpus material (theory and practice)".[2] In this project, two main goals were pursued: hands-on work with corpus data resulting in an explicit representation of cross-lingual meaning relations, and a theoretical comparative study particularly focused on differences between the Czech and English verbal valency structure. Theoretical aspects include both the description of verbal valency and the description of interlinking the translational verbal equivalents, focusing on comparison of the existing approaches in the two languages. This project is

[1] `http://lindat.mff.cuni.cz/services/CzEngVallex`

[2] A research grant supported by the Grant Agency of the Czech Republic under the id GP13-03351P

 Corresponding author: uresova@ufal.mff.cuni.cz
Cite as: Zdeňka Urešová, Eva Fučíková, Jana Šindlerová. CzEngVallex: a Bilingual Czech-English Valency Lexicon. The Prague Bulletin of Mathematical Linguistics No. 105, 2016, pp. 17–50. doi: 10.1515/pralin-2016-0001.

based on the Functional Generative Description Valency Theory (FGDVT) and on its application to a corpus, namely to the Prague Czech-English Dependency Treebank (PCEDT)[3] (Hajič et al., 2011). This theoretical approach is highly suitable for the proposed specification of relations of verbal valency frames in both languages. The work with the data includes the creation of a parallel Czech-English valency lexicon which is interlinked with real examples of valency usage in the broad context of the PCEDT.

The underlying idea of the project builds on the assumption that verbal valency is the core structural property of the clause, therefore, capturing the alignment of the translationally equivalent verbs, as well as the mappings[4] of their valency positions, should provide a valuable model of basic patterns within cross-lingual semantic relations. Moreover, such a resource that stores interlingual valency relations for several thousands of verbs and verb pairs might enable us making predictions (on the basis of semantic relatedness, or verb classes) about the verbs unseen in the text.

This article is structured as follows: after a theoretical background (Sec. 2) we present the basic structure of the CzEngVallex lexicon (Sec. 3, published in part in Urešová et al. (2015)). The annotation environment and process description follows (Sec. 4, Sec. 5). Linguistic issues related to the annotated data using CzEngVallex are described in Sec. 6 and in Sec. 7 (of which Sec. 7.1 to 7.3 have been published in part in Šindlerová et al. (2015)). We conclude with suggestions concerning possible applications and future work.

2. Theoretical background

Our approach to the issues of valency of Czech and English verbs applied in this project is based on the following points of view and uses the following principles and features (Sec. 2.1–2.2).

2.1. Valency in the FGD

The project draws on the Functional Generative Description Valency Theory. In this dependency approach, valency is seen as the property of some lexical items, verbs above all, to select for certain complementations in order to form larger units of meaning. The governing lexical unit then governs both the morphological properties of the dependent elements and their semantic interpretation (roles). The number and realization of the dependent elements constituting the valency structure of the phrase (or sentence) can be represented by valency frames, which can be listed in valency lexicons.

[3]http://hdl.handle.net/11858/00-097C-0000-0015-8DAF-4

[4]Here, we often use the terms “mapping” and “alignment” interchangeably. Though by “mapping”, we usually refer to the abstract notion of semantic equivalence of expressions between languages, and by “alignment”, we refer to its practical implementation in the data.

The basics of the FGDVT can be found, e.g., in Panevová (1974). The FGD approaches valency as a special relation between a governing word and its dependents.[5] This relation belongs to the level of deep syntax (tectogrammatical layer of linguistic description). It combines a syntactic and a semantic approach for distinguishing valency elements. The verb is considered to be the core of the sentence (or clause, as the case may be). The relation between the dependent and its governor at the tectogrammatical layer is represented by a *functor*, which is a label representing the semantic value of a syntactic dependency relation and expresses the function of the complementation in the clause. For a full list of all dependency relations and their labels, see Mikulová et al. (2006a).

The FGDVT works with a systematic classification of verbal valency complementations (arguments)[6] along two axes. The first axis represents the opposition between inner complementations (actants) and free complementations (adjuncts) and it is determined independently of any lexical unit. The other axis relates to the distinction between obligatory and optional complementations, for each verb sense separately.

There are five "inner participants" (actants) in the FGDVT: Actor/Bearer (ACT), Patient (PAT), Addressee (ADDR), Origin (ORIG) and Effect (EFF). Which functors are considered actants has been determined according to two criteria. The first one says that actants can occur at most once as a dependent of a single occurrence of a particular verb (excluding apposition and coordination). According to the second criterion, an actant is restricted to only a relatively closed class of verbs.

Out of the five actant types, the FGDVT states that the first two are connected with no specific globally defined semantics, contrary to the remaining three ones. The first actant is always the Actor (ACT), the second one is always the Patient (PAT). The Addressee (ADDR) is the semantic counterpart of an indirect object that serves as a recipient or simply an "addressee" of the event described by the verb. The Effect (EFF) is the semantic counterpart of the second indirect object describing typically the result of the event (or the contents of an indirect speech, for example, or a state as described by a verbal attribute). The Origin (ORIG) also comes as the second (or third or fourth) indirect object, describing the origin of the event (in the "creation" sense, such as *to build from metal sheets*.ORIG, not in the directional sense).

The FGDVT has further adopted the concept of shifting of "cognitive roles". According to this special rule, semantic Effect, semantic Addressee and/or semantic Origin are shifted to the Patient position in case the verb has only two actants. Similarly, any of the actant roles are shifted to the Actor position in case the verb has only a single valency position.

[5] For the sake of brevity, we will further refer only to the valency of verbs, since the CzEngVallex contains so far only the alignment of verb pairs.

[6] In the following sections, we will use the term "argument" for any of the complementations of a particular verb (sense) entry in the lexicon, i.e., for actants and adjuncts included in such a valency frame.

The repertory of adjuncts (free modifications) is much larger (about 50) than that of actants (see again Mikulová et al. (2006a)). Adjuncts are always determined semantically; their set is divided into several subclasses, such as temporal (TWHEN, TSIN, TTILL, TFL, TFHL, THO, TPAR, TFRWH, TOWH), local (LOC, DIR1, DIR2, DIR3), causal (such as CAUS for cause, AIM for purpose, CRIT for 'according to', etc.) and other free complementations (MANN for general 'manner', ACMP for accompaniment, EXT for extent, MEANS, INTF for intensifier, BEN for benefactor, etc.). Adjuncts may be seen as deep-layer counterparts of surface adverbial complementations. More adjuncts of the same type can occur as dependents on a particular occurrence of the verb and adjuncts may modify in principle any verb – this is also where their name ('free complementations') comes from. Unlike actants, morphemic realization of adjuncts is rarely (if ever) restricted by a particular verb.

Due to this "free nature" of adjuncts, only the presence of actants (obligatory or optional) and obligatory adjuncts is considered necessary in any verbal valency frame (the FGDVT is thus said to use the notion of valency in its "narrow" sense): optional adjuncts are (as a general rule) not listed in the valency frame. As mentioned above, both actants and adjuncts can be in their relation to a particular word either obligatory (that means obligatorily present at the tectogrammatical level) or optional (that means not necessarily present in any sentence where the verb is used). It must be said that this definition of obligatoriness and optionality does not cover surface deletions but only semantically necessary elements.

Since the surface appearance of a complementation does not really help to distinguish between obligatory and optional elements, other criteria must be used. Specifically, the 'dialogue test' is used. It is a method based on asking a question about the element that is supposed to be known to the speaker because it follows from the meaning of the verb: if the speaker can answer the hearer's follow-up wh-question about the given complementation with *I don't know* (without confusing the hearer), it means that the given complementation is semantically optional. On the other hand, if the answer *I don't know* is disruptive in the (assumed) conversation, then the given complementation is considered to be semantically obligatory. For further details, see Urešová (2011a).

2.2. Comparative character and corpus approach to cross-language research

We are interested in differences in the expression of the same contents in two typologically different languages, namely Czech and English. The initial hypothesis is that even in relatively literal or exact translation, where the information and the meaning the sentences carry in both languages is essentially the same–as exemplified in economic, news, and similar non-artistic genres–the core sentence structure (i.e., the main verb of a clause and its arguments) often differs due to intrinsic language differences. Comparing Czech and English valency frames and their arguments, based on their usage in a parallel corpus, is expected to enable not only the detection of the types of divergences of expression in the core sentence structure but also a quantitative analysis of their similarities and differences, thanks to the substantial size of the corpora available.

Both lexicons, which we used as a starting point, are based on the same theoretical foundations (cf. Sec. 2.1). Our task was thus slightly simplified in that we were not comparing two different valency theories, but rather an application of a single theoretical (and formal) framework to two particular languages (and to a translated, i.e., parallel corpus material). Such approach has, we believe, a major advantage: we are able to pinpoint the differences much more clearly against a unified theoretical background, as opposed to a possibly fuzzy picture which widely differing valency theories might give.

Our approach to the comparative study of valency builds on the growing role of computer corpora in linguistic research. Our study is based on corpus examples with natural contexts, which gives well-founded research results backed also by quantitative findings. Therefore, a detailed and thorough work with electronically created and accessible data, namely, with the PDT-Vallex and the EngVallex lexicons and the PCEDT, are the foundations we build our research on.

3. CzEngVallex reference data

For the CzEngVallex project, two treebanks are most relevant: the PDT[7] and the PCEDT[8] which contain manual annotation of morphology, syntax and tectogrammatics (semantics).

Next, we work with the PDT-Vallex verbal valency lexicon for Czech (Urešová, 2011b) and with a similar resource for English called EngVallex (Cinková, 2006).

These data resources are the "input" material for the creation of the CzEngVallex. Also, they are heavily referred to from the resulting CzEngVallex and can thus be considered an integral part of it.

3.1. Czech-English parallel corpus

The CzEngVallex primary data source is the parallel Prague Czech-English Dependency Treebank (PCEDT). The PCEDT is a sentence-parallel treebank based on the texts of the Wall Street Journal part of the Penn Treebank[9] and their manual (human) translations.

It is annotated on several layers, of which the tectogrammatical layer (layer of deep syntactic dependency relations) includes also the annotation of verbal valency relations. The tectogrammatical annotation of this corpus includes also links to two va-

[7] `http://ufal.mff.cuni.cz/pdt/`

[8] `https://catalog.ldc.upenn.edu/LDC2004T25`

[9] `https://catalog.ldc.upenn.edu/LDC99T42`

lency lexicons, the PDT-Vallex (for Czech) and the EngVallex (for English), see their detailed description below.

3.2. Czech and English valency lexicons

3.2.1. PDT-Vallex - Czech valency lexicon

The Czech valency lexicon, called PDT-Vallex,[10] is publicly available as a part of the one-million-word Prague Dependency Treebank (PDT) version 2 published by the Linguistic Data Consortium.[11] It has been developed as a resource for valency annotation in the PDT; for details, see Urešová (2011b). As such, it has been designed in close connection to the specification of the treebank annotation. The "bottom up", data-driven practical approach to the forming of the valency lexicon had made it possible for the first time to confront the already existing FGDVT and the real usage of language. Precise linking of each verb occurrence to the valency lexicon has made it possible to verify the information contained in the valency lexicon entry against the corpus by automatic means, making it a reliable resource for further research.

Each valency entry in the lexicon contains a headword, according to which the valency frames are grouped, indexed, and sorted. The valency frame contains the following specifications: the number of valency frame members, their labels, the obligatoriness feature and the surface form of valency frame members. Any concrete lexical realization of the particular valency frame is exemplified by an appropriate example, i.e., an understandable fragment of a Czech sentence, taken almost exclusively from the PDT. Notes help to delimit the meaning of the individual valency frames inside the valency entry. Typically, synonyms, antonyms and aspectual counterparts serve as notes. For a detailed information about the actual structure of the PDT-Vallex entry, see Urešová (2011a).

The version of the PDT-Vallex used for the CzEngVallex contains 11,933 valency frames for 7,121 verbs. The verbs and frames come mostly from the data appearing in the PDT, version 2.0, and the PCEDT, version 2.0. The lexicon is being constantly enlarged with data provided by further annotations.

3.2.2. EngVallex - English valency Lexicon

The EngVallex[12] is a lexicon of English verbs, also built on the grounds of the FGDVT. It was created by a (largely manual) adaptation of an already existing resource for English with similar purpose, namely the PropBank Lexicon (Palmer et al., 2005; Kingsbury and Palmer, 2002), to the PDT labeling standards (see also Cinková (2006)). During the adaptation process, arguments were re-labeled, obligatoriness was marked

[10] http://hdl.handle.net/11858/00-097C-0000-0023-4338-F

[11] http://www.ldc.upenn.edu/LDC2006T01

[12] http://hdl.handle.net/11858/00-097C-0000-0023-4337-2

for each valency slot, frames with identical meaning were unified and sometimes, frames with a too general meaning were split. Links to PropBank frames have been preserved wherever possible. The EngVallex was used for the valency annotation of the Wall Street Journal part of the Penn Treebank during its manual annotation on the tectogrammatical layer; the result is the English side of the PCEDT.

The EngVallex currently contains 7,148 valency frames for 4,337 verbs.

4. Building CzEngVallex

4.1. The annotation goal

To meet the goals stated in Sec. 1, an explicit linking between valency frames of Czech and English verbs based on a parallel corpus is needed. This has been accomplished by creating the bilingual Czech-English Valency Lexicon (CzEngVallex).[13]

The CzEngVallex stores alignments between Czech and English valency frames and their arguments. The resulting alignments are captured in a stand-off mode (in a file called `frames_pairs.xml`). This file is the "entry point" to the CzEngVallex; it cannot be used independently, since it refers to the valency frame descriptions contained in both the PDT-Vallex and the EngVallex, and it also relies on the PCEDT as the underlying corpus.

The idea of CzEngVallex builds on Šindlerová and Bojar (2009) and Bojar and Šindlerová (2010). However, only a pilot experiment has been described in these two papers; the actual process of creating CzEngVallex differed from suggestions in these papers in several substantial aspects.

4.2. CzEngVallex structure

The CzEngVallex builds on all the resources mentioned in Sec. 3. It is technically a single XML file `frames_pairs.xml` (shown in Fig. 1) which lists for each included English verb (identified by a verb `id`) a list of its valency frames (identified by a valency frame `id`), and for each English valency frame all the collected frames-pairs, and for each of the collected frames-pairs (identified by a pair `id`) the pairings of their valency slots (identified by functors).

Aligned pairs of individual verb frames are grouped by the English verb frame (<en_frame>) (cf. Fig. 1), and for each English verb sense, their Czech counterparts are listed (<frame_pair>). For each of such pairs, all the aligned valency slots are listed and referred to by the functor assigned to the slot in the respective valency lexicon (the PDT-Vallex for Czech, the EngVallex for English).

[13] Available for browsing and searching at `http://lindat.mff.cuni.cz/services/CzEngVallex`, download from `https://lindat.mff.cuni.cz/repository/xmlui/handle/11234/1-1512`

```
<frames_pairs owner="...">
 <head>...</head>
 <body>
  <valency_word id="vw1484" vw_id="ev-w1869">
   <en_frame id="vw1484f1" en_id="ev-w1869f1">
    ...
    <frame_pair id="vw1484f1p8" cs_id="v-w8735f1">
     <slots>
      <slot en_functor="ACT" cs_functor="ACT"/>
      <slot en_functor="PAT" cs_functor="PAT"/>
      <slot en_functor="EFF" cs_functor="---"/>
     </slots>
    </frame_pair>
    ...
   </en_frame>
  </valency_word>
 </body>
</frames_pairs>
```

Figure 1. Structure of the CzEngVallex (part of limit pairing)

In the example in Fig. 1, for the pair *limit*[14] - *zabránit* (lit. *limit/prevent*) we can observe a match of the first two actants (ACT:ACT, PAT:PAT) and a zero alignment (cf. Sec. 6.2.2) of the third frame element: EFF,[15] which does not match any verb argument for this particular Czech counterpart.

It is crucial to mention here that while all verb–verb pairs have been aligned, annotated and then collected in this pairing lexicon, there are also many verb–non-verb or non-verb–verb pairs, which have been left aside for the first version of the CzEngVallex, since none of the underlying lexicons has enough entries covering nominal valency included.

5. Annotation environment

5.1. Prerequisites

The annotation was done over the bilingual data from the parallel PCEDT 2.0.[16] The annotation interface for building the CzEngVallex was constructed as an extension of the tree editor TrEd (Pajas and Fabian, 2011)[17] environment.

[14]Frame ID ev-w1869f1, which has been created from limit.01 in the PropBank, as in *... which.*ACT *limits any individual holding.*PAT *to 15%.*EFF

[15]Marked as optional in EngVallex but optional actants must still be aligned.

[16]http://ufal.mff.cuni.cz/pcedt2.0/en/index.html

[17]http://ufal.mff.cuni.cz/tred

TrEd is a fully customizable and programmable graphical editor and viewer for tree-like structures. Among other projects, it was used as the main annotation tool for the tectogrammatical annotation of both source treebanks (PDT and PCEDT). It allows displaying and annotating sentential tree structures on multiple linguistic layers with a variety of tags using either the Prague Markup Language (PML) format[18] or the Treex format.[19]

Treex (formerly TectoMT) (Žabokrtský, 2011; Popel and Žabokrtský, 2010) is a development framework for general as well as specialized NLP tasks (such as machine translation) working with many representations of text and sentence structure, including tectogrammatically annotated structures. It offers its own file format, which is capable of storing and displaying (using TrEd) multiple tree structures at once, hence it is a fitting environment when cross-lingual relations are involved.

We have tried to keep the annotation environment as simple and transparent as possible, though still leaving all its important features available (see Fig. 2). It provides an annotation mode for valency frames alignment between the PDT-Vallex and the EngVallex. This extension builds on previously used TrEd extensions: the pdt2.0 extension (for the annotation of the PDT 2.0), the PDT-Vallex extension, and the pedt extension (for annotating the English side of the PCEDT); all these extensions offer functions necessary for browsing Czech and English treebanks and their valency lexicons, while the CzEngVallex extension itself provides the cross-lingual interlinking function.

5.2. Preprocessing and data preparation

The following steps were taken before the start of the annotation proper:

- automatic alignment on the word level of the PCEDT 2.0;
- preliminary collection of all verb-verb alignments and alignments of their complementations based on the referred-to valency lexicon entries, as they had been included in the PCEDT;
- preparation of lists grouping together all verb-sense pairs for every English verb as collected within the previous step.[20]

For the word alignment of the PCEDT data, the GIZA++[21] algorithm was used, and subsequently, this alignment was mapped to the nodes of the corresponding (deep/tectogrammatical) dependency trees representing the original and the translated sentence.

[18] `http://ufal.mff.cuni.cz/jazz/PML`

[19] `http://ufal.mff.cuni.cz/treex`

[20] These lists of verb occurrences in the parallel treebank are technically called 'filelists'.

[21] `https://code.google.com/p/giza-pp`

The resulting pairs were grouped by these references, one group for each English verb, and stored as *filelists*, which can be fed directly into the annotation tool TrEd (described in Sec. 5.4). Thus, the annotator was able to inspect the same verb occurrences together in a single data block. Similarly, the individual pairs for the same source verb sense were sorted in succession within the groups. The process of correcting, re-aligning (when necessary) and finally collecting the verb–verb alignments followed, based on the EngVallex and the PDT-Vallex references contained already in the treebank data for both translation sides.

5.3. The filelists

The corresponding pairs of Czech and English verbs were looked up in the PCEDT, using a btred[22] script. The script searches through the alignment attribute of the English verb nodes, where the information about the connection to the Czech counterpart is usually stored. All instances of individual verb pairs in the PCEDT were then listed in the form of filelists containing treebank position identifiers of the corresponding nodes. As such, they can be browsed alphabetically, or on the basis of pair frequency in a treebank, or employing other useful criteria.

Filelists were sorted by the English verb lemma and organized alphabetically into folders according to the first letter of the source verb. If a single English verb corresponded to more than one Czech verb, those verbs were placed in the same folder - the name of the folder then consists of the name of the English verb, the number of corresponding Czech verbs and the number of occurrences in the parallel corpus (e.g., *abate.3v.4p*). The filelists' names were designed according to the following rules:

(i) if there exist more Czech verbs to a given English verb in the parallel corpus, the filelist corresponding to one of the pairs will be placed in a directory named after the English verb, and will bear a name containing the Czech verb and the number of occurrences of this pair in the parallel corpus (e.g., for the pair *abate-polevit*, a filelist named *polevit.2.fl* is in a directory *abate.3v.4p*);

(ii) if there exists only a single Czech verb to a given English verb in the parallel corpus, the name of the filelist for this pair will contain both the English and Czech verb and the number of occurrences of this pair in the parallel corpus (e.g., *abide_by.1v.2p.dodržovat.2.fl*).

The annotator received a set of all available sentences for each verb pair at once. In total, there were 92,889 sentences, which were split into 15,931 filelists with an average number of sentences in one filelist 5,83 (median 1). The most frequent pair is *be→být*, which has 10,287 instances in its filelist.

Single-instance filelists[23] have been, for the sake of annotation efficiency, unified into a single filelist within the corresponding folder, e.g., for the verb *abate* the filelists

[22] `http://ufal.mff.cuni.cz/pdt2.0/doc/tools/tred/bn-tutorial.html`

[23] By single-instance filelists we mean verb pairs with only a single occurrence in the parallel corpus.

zmírnit.1.fl and *zmírnit_se.1.fl* merge into one filelist *abate.1_1.2.fl*; similarly, the filelists *abdicate.1v.1p.zbavovat_se.1.fl*, *abet.1v.1p.podporovat.1.fl*, *abort.1v.1p.potratit.1.fl* etc. are absorbed in a single filelist *a.1_1.30.fl*).

The annotators thus eventually processed 7,891 filelist in total, with the average number of sentences in the filelist 11,77 (median 3).[24]

5.4. The annotation process

During the actual annotation process, English and Czech verbs and their arguments were manually aligned or re-aligned, and after checking carefully all the occurrences of any given pair in the PCEDT data, the corresponding arguments were captured in the CzEngVallex lexicon, using the structure described in Sec. 4.2.

Even though all PCEDT occurrences of all verb–verb pairs were inspected manually, the process was helped substantially by several automatic preprocessing steps, as described in Sec. 5.2.

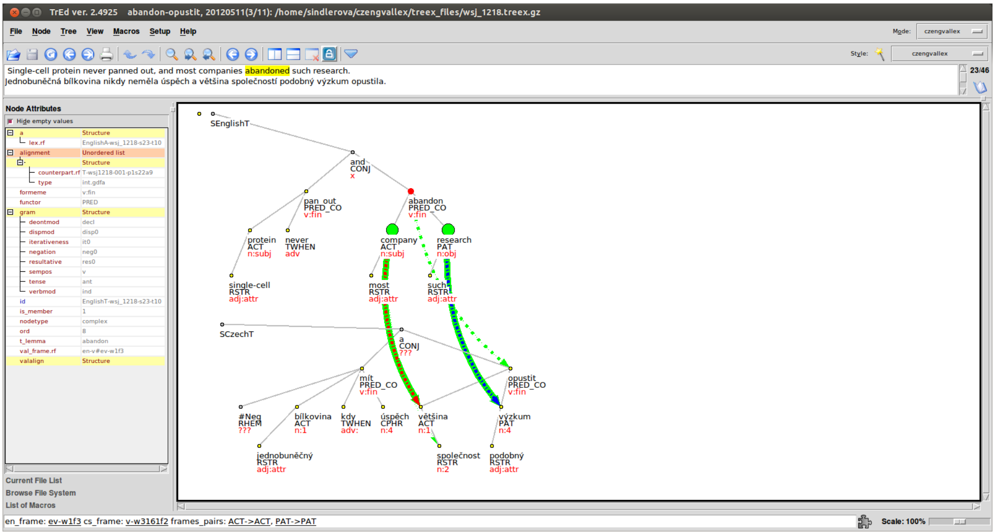


Figure 2. Annotation environment at work

[24]For detailed work with filelists see Urešová et al. (2015).

5.5. Manual alignment - the starting point

The environment described in Sec. 5 was used to display, edit, collect, and store the alignments between Czech and English valency frames.

Each annotator had her/his own copy of the PDT-Vallex, the EngVallex and the PCEDT and the filelists to work on (Sec. 5.2).[25]

S/he was expected to go through all verb occurrences in the filelist and build a typical valency frame alignment for each verb sense. S/he was also expected to deal with the potential conflicting cases (choose the most probable alignment option, mark complicated issues, such as missing or inappropriate frames or wrong tree structure in a note, etc.). Once collected, the frame alignment was automatically extended to all occurrences of the pair of the valency frames; it was the annotator's responsibility to check all the occurrences of such a pair if they correspond to the collected alignment, as recorded in the CzEngVallex.

Direct changes (changing the tree structure or frame adjustments) in the treebank were disallowed, though the extension allowed storing some minor type of changes (change of functor label) in specific CzEngVallex-related attributes. Also, the annotator reported problems through a note system for later corrections,[26] and s/he was allowed to change the valency frame link if considered inappropriate.

6. Understanding CzEngVallex

While this paper is not a substitute for the annotation guidelines, the basic rules for aligning verbs and their arguments will be described here so that the reader can understand the CzEngVallex data - what was annotated, what was not, in which cases examples were not included, treatment of convention differences in both valency lexicons, and more.

All details regarding annotation guidelines, annotation workflow and functionality of the annotation extension of TrEd are given in the CzEngVallex Technical Report (Urešová et al., 2015).

6.1. Verb pairs to include (or exclude)

As explained previously, CzEngVallex contains only those verb pairs for which a reasonable alignment was found in the treebank; sometimes, all occurrences (one or more) of the same frame pair align such diverging structures that they could not be aligned.

These cases include:

[25] A subversion system has been used for easy synchronization between annotators' laptops and the main data store.

[26] The CzEngVallex extension offers specific pre-defined "note" attributes to the annotator, which can be extended by free text, cf. Urešová et al. (2015).

1. good translation but with too different syntax which can be the result of
   (a) the use of a language-specific syntactic structure,
   (b) translation of a single verb by multiple verbs and consequent untypical argument distribution between these verbs;
2. semantically incorrect or too loose translation resulting in a syntactic difference.

Judging the degree of syntactic diversity has been fully up to the annotator. In case of complex and rare syntactic differences, the annotator was required not to include the sentence (or more sentences for a given frame pair) in the annotation. The reason for omission is usually described in the note attribute. For example, if the translation was substantially inaccurate or if the translation was too loose, the sentences remained manually "unannotated," i.e., there was no attempt to correct alignments in the data or to make other data adjustments. The annotator was required to leave a note saying, e.g., "too loose translation".

In case all occurrences of a verb pair were deemed unalignable, such a verb pair is not included in the `frames_pairs.xml` file.

6.2. Discrepancies and conflicts in annotation

Ideally, each pair of frames is supposed to have only a single way of argument alignments. This follows from the semantic character of the tectogrammatical structure. Due to the deep character of the description, it is also supposed that the alignment should be to a great extent "parallel," i.e., that the nodes of the two trees ideally correspond 1:1 and that their functors match.

Nevertheless, this is often not the case. There are discrepancies and conflicts of different kinds in the data, as the CzEngVallex annotation reflects.

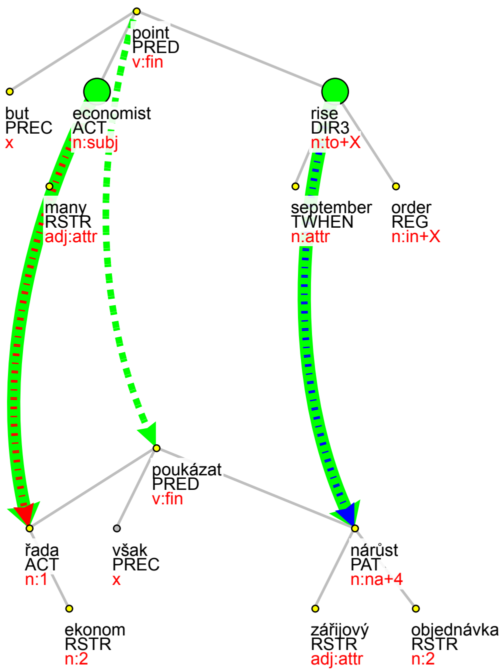
By discrepancies, we refer either to the so-called zero alignment (see Sec. 6.2.2), i.e., places where an argument node in one of the languages is translated in such a way that it is not a direct dependent (i.e., not an argument) of the aligned verb in the other language, or to the functor mismatch (6.2.1), i.e., when two aligned nodes have different tectogrammatical functor labels.

By conflicts in annotation (Sec. 6.2.3), we refer to cases where the alignment of the verb or its arguments looks differently in different sentences in the corpus. In other words, for that frame pair, one such alignment would be in conflict with another alignment observed elsewhere in the data.[27]

[27]The design of CzEngVallex (Sec. 4.2), as mirrored in the structure of the `frames_pairs.xml` file, does not allow for alternative argument alignments for the same verb frame pair. Please recall that verb frames already represent a single verb sense, thus this type of conflict should not be blamed on potentially mixed senses of the verb involved.

6.2.1. Functor mismatch

By functor mismatch, we mean alignment of nodes with different functor labels (see example in Fig. 3).[28] These alignments can involve either (proper) actant-actant mapping, or even an actant-adjunct mapping. The causes for functor mismatch often involve different morphosyntactic realization which was treated differently in the two languages, rather than a clear semantic difference.



En: But many economists pointed to a ... September rise in orders ...
Cz: Řada ekonomů však poukázala na ... zářijový nárůst objednávek, ...

Figure 3. Functor mismatch DIR3⟶PAT *in the data*

Though this is in most cases technically unproblematic, we provide some notes of the common causes of functor mismatch in the following paragraphs.

[28]In the examples displayed, the green lines connect either the annotated verb pair or the already collected argument pairs, the automatic node alignment suggestion is displayed as a blue arrow, the manually corrected alignment is marked as a red arrow. The images have been cropped or otherwise adjusted for the sake of clarity.

The data show that it is quite often the case that the alignment connects an actant (usually on the English side) to an adjunct (usually on the Czech side), for example ADDR to DIR3 or LOC, also EFF to COMPL, ACT to LOC, ACT to CAUS etc. These differences often have grounds in different morphosyntactic forms of the given modifications, which was taken as decisive for using an adjunct instead of an actant (mostly on the Czech side due to its richer morphology). This is a feature of the underlying linguistic theory that was perhaps a bit overstressed in the original treebank (PDT) annotation when assigning the functor(s) to slots in the valency frames.

Since the morphosyntactic forms of the valency complementations are to a great extent fixed with the given verb, the alignment for individual functor pairs seems to be quite consistent throughout certain verb pairs or even verb classes.[29] For example, (English) ADDR to (Czech) DIR3 appears with, e.g., the verbs *commit/svěřit* (En: *...committing more than half their funds to either.*ADDR *of those alternatives* / Cz: *...svěřilo více než polovinu svých prostředků do jediné.*DIR3 *z těchto alternativ*). Similarly, the link (English) EFF to (Czech) COMPL appears with the verb pair *consider/posoudit* (En: *...will be considered timely.*EFF *if postmarked no later than Sunday* / Cz: *...budou posouzeny jako včas podané nabídky.*COMPL).

This kind of functor mismatch can occur with any actant label, even with the ACT. For example, the case of ACT aligning to MEANS appears due to a known problem of the so-called instrument-subject alternation, here illustrated with the verb pair *please/potěšit*: En: *Pemex's customers are pleased with the company's new spirit.*MEANS / Cz: *Zákazníky společnosti Pemex rovněž potěšil nový elán.*ACT *společnosti*.

In case there is a "third" actant in the structure, this third (or higher-numbered) actant may also differ in labeling in English and Czech, even in cases where the semantic correspondence is clear. For example, see the following occurrence of the verb pair *insulate/chránit*: En: *...will further insulate them.*PAT *from the destructive effects.*ORIG / Cz: *...je.*PAT *bude dále chránit před destruktivními vlivy.*EFF. Here, the English ORIG corresponds to the Czech EFF. While this is not a technical problem, it signals unclear definitions of those actant labels in the Czech and English guidelines for valency entries. This deficiency was found both for actants, semantically close adjuncts and for actant/adjunct pairs, e.g., EFF/MEANS mapping: for the verb pair *outfit/vybavovat*: En: *…will outfit every computer with a hard drive.*EFF / Cz: *...bude vybavovat všechny počítače pevným diskem.*MEANS. The question of labeling the actants (PAT ORIG x ADDR PAT) arose also in the following example for the verb pair *rid/zbavit*: En: *...to clean up Boston Harbor or rid their beaches.*PAT *of medical waste.*ORIG / Cz: *...zbavit pláže.*ADDR *nemocničního odpadu.*PAT.

An example of semantically close functors mismatch is the problem of a "dynamic versus static expression of location", i.e., DIR3/LOC mismatch: for the verb pair *include/zahrnout*, the data offer the following example: En: *...real-estate assets are in-*

[29] At this time, we have not fully investigated this interesting issue in a quantitative way, leaving it for future research.

cluded in the capital-gains provision.DIR3 / Cz.: *…nemovitý majetek je v ustanovení*.LOC *o kapitálových ziscích zahrnut*; or: En: *...prime minister ordered to deposit 57 million in bank*.LOC / Cz: *…ministerský předseda nařídil uložit asi 57 milionů dolarů do banky*.DIR3. Note that the theory based on deep syntactic frames does not allow to reinterpret labels in semantic changes caused by syntactic shifts such as passivization.

The fact that the functor mismatch often occurs when semantically parallel structures differ in morphological realization only, and in some cases even allow alternative interpretation, leads us to the need to reconsider the valency slot labeling schemes for both English and Czech, and more precisely define the “semantics” of these labeling schemes, since often the differences in argument and/or adjunct labels do not seem warranted.

6.2.2. Zero alignment

By zero alignment we mean such structural configurations that involve different number of arguments in the corresponding syntactic structures, i.e., an alignment of “something” on one side of the translation to “nothing” on the other side. There are various reasons for zero alignment, e.g., a simple absence of a lexical or structural counterpart in the translation, or deeper embedding of an argument counterpart in a subtree.

In Fig. 4, the reason is that in English the word *earnings* is treated as an argument of the light verb *have*, whereas in Czech its counterpart (*výdělky*) depends on the nominal part of the light verb constructions (the word *dopad* - lit. *impact*).

A slightly different case appears for the verb pair *call/volat*, En: *...this calls into question the validity of the R... theory* / Cz: *...to volá po otázce po správnosti R... teorie*: the Czech equivalent *správnost* to the English *validity*.PATient is embedded, since the English construction is considered an idiom (*calls into question*), marking *into question* as DPHR. In Czech, *správnost* carries the RSTR label and depends not on the verb, but on the noun *otázka* (lit. *question*).

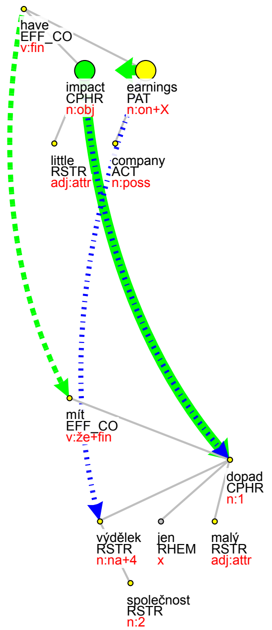
The usual way of treating zero alignment is keeping the alignment of the appropriate “superfluous” node to “no specific node”.

Zero alignment is caused, i.a., systematically by certain linguistic phenomena, such as different complexity of verbal meaning expression or loose or specific translation. Some of the cases are treated in Sec. 7.1 to 7.3.

6.2.3. Conflicts

Conflicts, as defined above, arise if the verb argument annotation at one place in the data is inconsistent with another occurrence in the data.

First, there may be problems with the granularity of verb senses as represented by the verb frames in the PDT-Vallex and EngVallex lexicons, which is then displayed in the aligned PCEDT data (as opposed to the Czech and English sides when taken separately, where it cannot be seen easily). With some verbs, the alignment as displayed



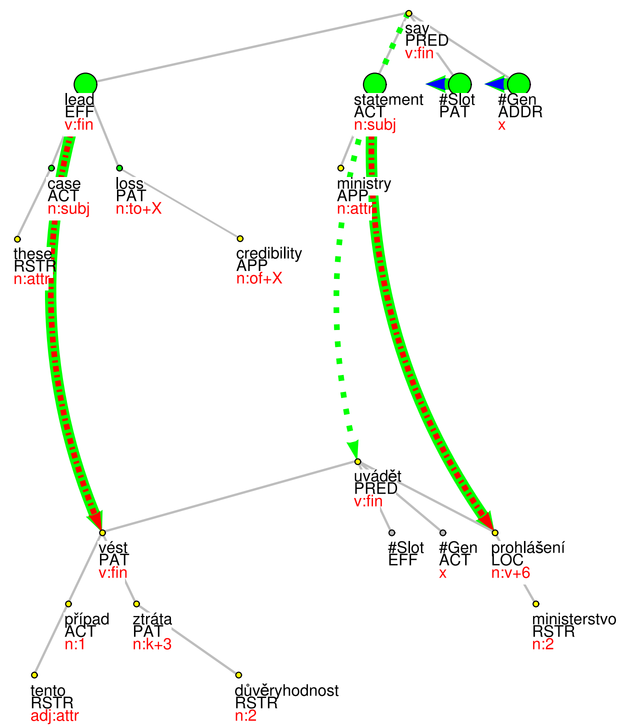
En: ... have little impact on the company's earnings.
Cz: ... bude mít na výdělky společnosti jen malý dopad.

Figure 4. Zero alignment (embedded argument) PAT⟶- - -

in the parallel data might show that two separate frames for two separate verb senses are needed, instead of the currently used one frame for both (or more), often due to certain overgeneralization in either of the lexicons. That is, the parallel data give a reason for more fine-grained distinctions in verb senses (i.e., more verb frames) for that particular verb in that valency lexicon.

For example, the English verb *bite* when translated as *kousnout* generates a conflict in the data. In one, rather idiomatic, occurrence, *bite one's lip*.PAT is translated with *kousnout se*.PAT *do rtu*.DIR3, thus aligning the English PAT with a Czech DIR3 functor. In another occurrence, arguably the more general one, the PAT actants of the verbs on both sides are aligned. Thus the data give evidence of a possible need of establishing a new frame for certain (for example, idiomatic) uses of the verb.

Second, conflicts arise in rather specific syntactic constructions, i.e., for two syntactic constructions, a default one and a specific one, which are otherwise considered to represent the same valency frame, though having a different placement of semantic modifications in the syntactic structure.



En: "These cases lead to the loss of ... credibility," a ministry statement said.
Cz: "Tyto případy vedou ke ztrátě důvěryhodnosti ...," uvádělo se v prohlášení ministerstva.

Figure 5. Conflicting occurrence of an ACT→LOC *alignment (vs.* ACT→ACT*)*

An example documenting this case is shown in Fig. 5, where we see a conflicting alignment for the pair *say–uvádět* (in the appropriate senses). In many (other) instances, the standard alignment of ACT (ACT→ACT) applies (*The president*.ACT *said that ...–Prezident*.ACT *uváděl, že ...*). However, in the parallel sentences depicted in Fig. 5: the same frame pair would lead to a different, non-identical mapping (ACT→LOC). This locative representation of the *medium of information transfer* modification (Cz: *prohlášení*), combined with a reflexive passive of the verb, is a syntactically typical alternation for Czech (but *only* for such a "medium" class of words, as opposed to persons etc.), whereas in English, the *medium* (En: *statement*) usually takes the subject (ACT in a canonical active sentence form) position in the sentence.

Third, conflicts can be lexically motivated, depending on the translation variant chosen by the translator. This differs from the first case above in that it is not possible to classify this as a difference in granularity of the valency frame(s), since the expression(s) used may not be considered clear idioms.

Conflicts have not been resolved on solid theoretical grounds in the current version of CzEngVallex, but notes from the annotation process have been preserved internally to reflect in future releases of the underlying treebanks, valency lexicons, or both (and, consequently, in CzEngVallex itself).

7. Specific linguistic issues

In the following sections, we describe some specific linguistic issues found in the data, we comment on their linguistic background and on the way they are annotated.

7.1. Catenative and modal verbs

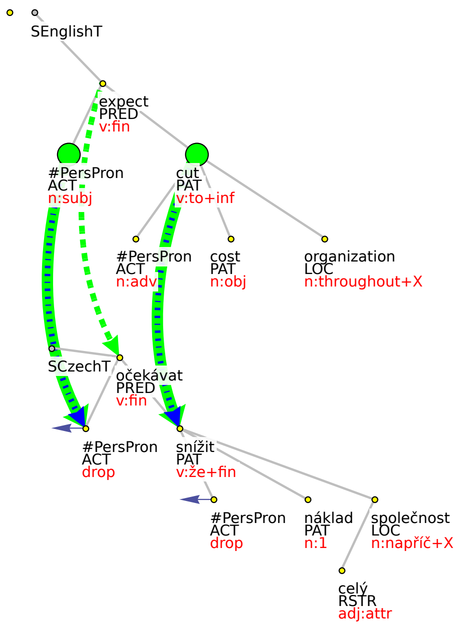
Special attention in the annotation was paid to verbs that form, together with another verb, a single homogeneous verb phrase, i.e., they precede another verb and function either as a chain element (catenative) or as an auxiliary (modal) verb. Catenative verbs are usually defined as those combining with non-finite verbal forms, with or without an intervening NP that might be interpreted as the subject of the dependent verbal form. Most of the classes described in Palmer (1974); Mindt (1999) can premodify main verbs and occupy the same syntactic position as auxiliaries or modals. They often cause some kind of structural discrepancy in the data.[30]

7.1.1. ECM Constructions, Raising to Object

Most Czech linguistic approaches do not recognize the term Exceptional Case Marking (ECM) in the sense of "raising to object", instead they generally address similar constructions under the label "accusative with infinitive". The difference between ECM and control verbs is not being taken into account in most of Czech grammars. In short, raising and ECM are generally considered a marginal phenomenon in Czech and are not being treated conceptually (Panevová, 1996), except for several attempts to describe agreement issues, e.g., the morphological behaviour of predicative complements described in a phrase structure grammar formalism (Przepiórkowski and Rosen, 2005).

The reason for this particular approach to ECM is probably rooted in the low frequency of ECM constructions in Czech. Czech sentences corresponding to English sentences with ECM mostly do not allow catenative constructions. They usually involve a standard dependent clause with a finite verb, see Fig.6, or they include a nominalization, thus keeping the structures strictly parallel.

[30]By a structural discrepancy in dependencies, we mean such structural configurations that involve different number of dependencies in the corresponding syntactic structures, i.e., an alignment of "something" on one side of the translation to "nothing" on the other side, see also Sec. 6.2.



En: They expect him to cut costs...
Cz: Očekávají, že sníží náklady...

Figure 6. Alignment of the ECM construction

The only exception are verbs of perception (*see*, *hear*), which usually allow both ways of Czech translation – with an accusative NP followed by a non-finite verb form (1a), or with a dependent clause (1b), not speaking about the third possibility involving an accusative NP followed by a dependent clause (1c).

(1) He saw Peter coming.

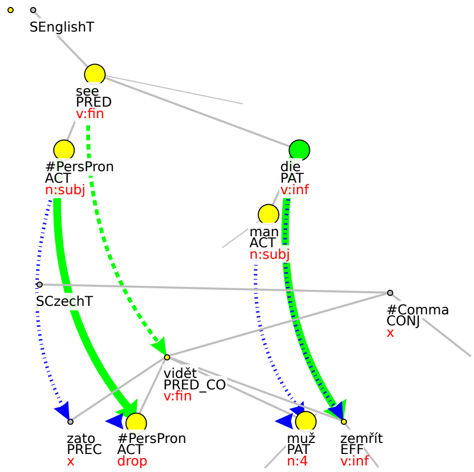
a. Viděl Petra přicházet.
He saw Peter.ACC to come.

b. Viděl, že Petr přichází.
He saw that Peter.NOM is coming.

c. Viděl Petra, jak přichází.
He saw Peter.ACC, how is coming.

In this type of accusative-infinitive sequence, the accusative element is in FGDVT analysed consistently as the direct object of the matrix verb (PAT) and the non-finite verb form then as the predicative complement of the verb (EFF).

The PCEDT annotation of verbs of perception is shown in Fig. 7, with frame arguments mapped in the following way: ACT→ACT; PAT→EFF; - - -→PAT. The corresponding arguments man-muž are interpreted as belonging to verbs in different levels of the structure.



En: I have seen [one or two] men die...
Cz: Zato jsem viděla [jednoho nebo dva] muže zemřít...

Figure 7. Alignment of the perception verbs' arguments.

The literature mentions two ways of ECM structural analysis, a flat one, representing the NP as dependent on the matrix verb, and a layered one, representing the intervening NP as the subject of the dependent verb. This mirrors the opinion that verbs allowing ECM usually have three syntactic, but only two semantic arguments. The practical solution is then a matter of decision between a syntactic and semantic approach to tree construction.

The English part of the PCEDT data was annotated in the layered manner,[31] thus most of the pairs in the treebank appear as strictly parallel. The consistency of structures is one of the most important advantages of the layered approach; there is no need of having two distinct valency frames for the two syntactic constructions of the verb, therefore, the semantic relatedness of the verb forms is kept.

[31] The annotation followed the original phrasal annotation of the data in the Penn Treebank.

On the other hand, the Czech part of the PCEDT data uses flat annotation, partly because the catenative construction with raising structure is fairly uncommon in Czech (cf. Sect. 7.1.1). The flat structure is easier to interpret, or translate in a morphologically correct way to the surface realization, but it requires multiple frames for semantically similar verb forms (the instances of the verb *to see* in *see the house fall* and *see the house* are in the FGD valency approach considered two distinct lexical units) and it also leaves alignment mismatches in the parallel data.

The treatment of ECM constructions in English and in Czech is different. It reflects both the differences internal to the languages and their consequences in theoretical thinking. Contrary to English, Czech nouns carry strong indicators of morphology – case, number and gender. The rules for the subject-verb agreement block overt realization of subjects of the infinitives. The accusative ending naturally leads to the interpretation of the presumed subject of the infinitive as the object of the matrix verb. The morphosyntactic representation is taken as a strong argument for using a flat structure in the semantic representation, and a covert co-referential element for filling the "empty" ACTor position of the infinitive. In English, in general, there is no such strong indication and therefore the layered structure is preferred in the semantic representation.

7.1.2. Object control verbs, equi verbs, causatives

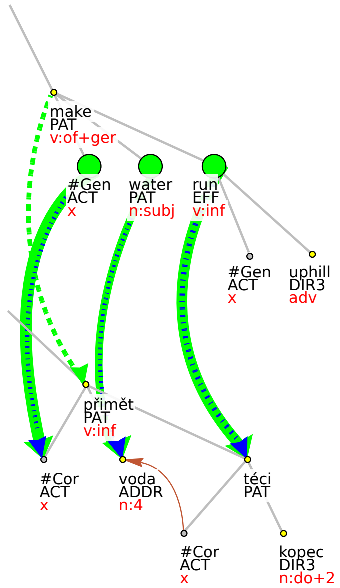
Contrary to the ECM constructions, object control verbs constructions (OCV), involving verbs such as *make, cause, or get*, are analyzed strictly as double-object in both languages. OCV constructions are similarly frequent in Czech and English and their alignment in the PCEDT data is balanced, see Fig. 8.[32]

Interestingly, it is sometimes the case that English control verbs in the treebank are translated with non-control, non-catenative verbs on the Czech side, and the intervening noun phrase is transformed to a dependent of the lower verb of the dependent clause (see Fig. 9).

The verb involved in this kind of translation shift may be either a more remote synonym, or a conversive verb.[33] Such a translation shift brings about (at least a slight) semantic shift in the interpretation, usually in the sense of de-causativisation of the meaning (*prompt→lead to*). of (any) language to suppress certain aspects of meaning without losing the general sense of synonymity.

[32]In Fig. 8, English ACT of *run* does not show the coreference link to *water* since the annotation of coreferential relations has not yet been completed on the English side of the PCEDT, as opposed to the Czech side (cf. the coreference link from ACT of *téci* to *voda*).

[33]Semantic conversion in our understanding relates different lexical units, or different meanings of the same lexical unit, which share the same situational meaning. The valency frames of conversive verbs can differ in the number and type of valency complementations, their obligatoriness or morphemic forms. Prototypically, semantic conversion involves permutation of situational modifications.



En: ...making water run uphill...
Cz: ...přimět vodu téct do kopce...

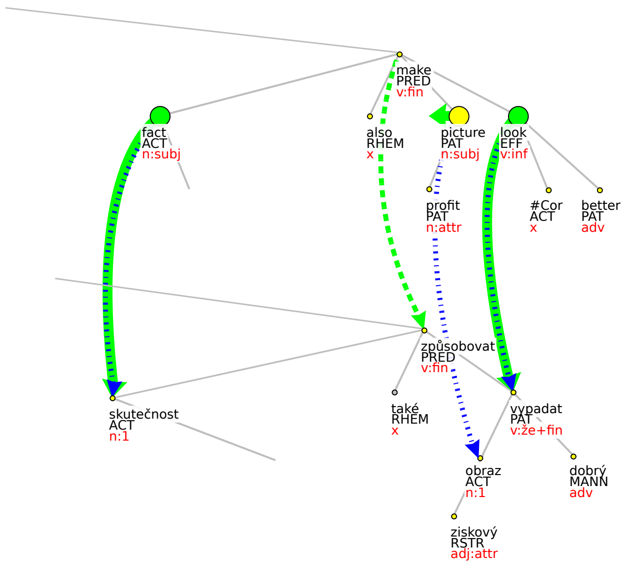
Figure 8. Alignment of the control verbs' arguments

Such occurrences have been treated as typical examples of zero alignment (see Sec. 6.2.2).

7.2. Complex Predication

By "complex predication" we mean a combination of two lexical units, usually a (semantically empty, or "light") verb and a noun (carrying main lexical meaning and marked with `CPHR` functor in the data), forming a predicate with a single semantic reference, e.g., *to make an announcement*, *to undertake preparations*, *to get an order*. There are some direct consequences for the syntactically annotated parallel data where we encounter two types of zero alignment.

First type of zero alignment is connected to the fact that a complex predication in one language can be easily translated with a one-word reference, and consequently aligned to a one-word predication, in the other language. This is quite a trivial case. In the data, then, one component of the complex predication remains unaligned. There



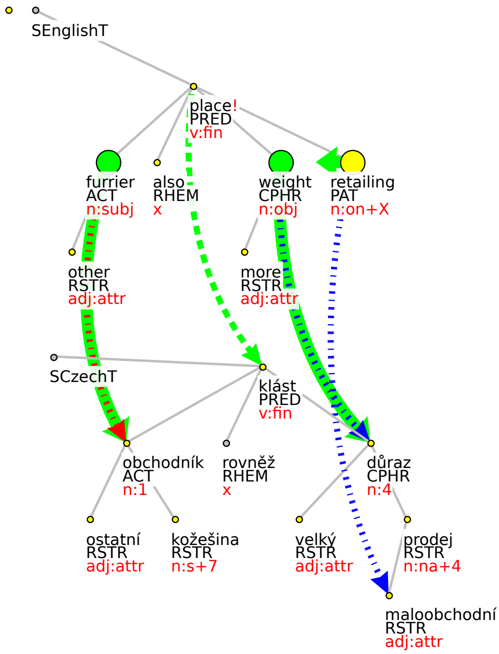
En: The fact ... will also make the profit picture look better.
Cz: Skutečnost ... způsobuje, že ziskový obraz vypadá lépe.

Figure 9. Alignment of English OCV with Czech non-OCV construction

are basically two ways of resolving such cases: either one can align the light verb with the full verb in the other language, or one can align the full verb with the dependent noun in the complex predication, based on the similarity of semantic content. In the CzEngVallex, the decision was to align the verbs, reflecting the fact that the verb and the noun phrase form a single unit from the semantic point of view.

The second type of zero alignment is connected to the presence of a "third" element within the complex predication structure, structured as dependent on the verb on one side, and on the predicative noun on the other side of the translation, e.g., En: *placed weight on retailing* - Cz: *klást důraz na prodej*, see Fig. 10.

Complex predicates have been annotated according to quite a complicated set of rules on the Czech side of the PCEDT data (Mikulová et al., 2006b). Those rules include also the so-called dual function of a valency complementation. There are two possible dependency positions for the "third" argument of the complex predicate: either it is modelled as the dependent of the semantically empty verb, or as a dependent of the nominal component. The decision between the two positions relies on multiple factors, such as valency structure of the semantically full use of the verb, valency



En: Other furriers have also placed more weight on retailing.
Cz: Ostatní obchodníci s kožešinami rovněž kladou větší důraz na maloobchodní prodej.

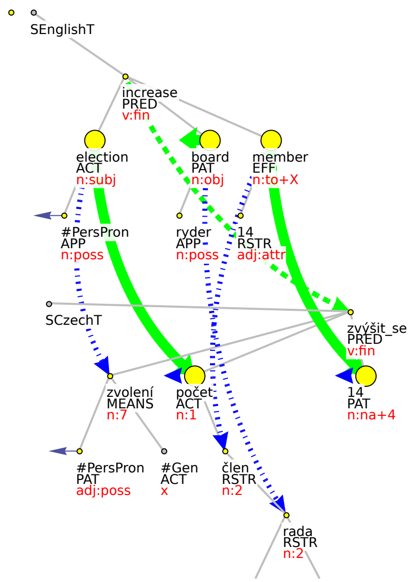
Figure 10. Mismatch due to complex predication solution

structure of the noun in other contexts, behaviour of synonymous verbs etc. On the Czech side, the "third" argument was strongly preferred to be a dependent of the nominal component. On the English side of the PCEDT, the preferred decision was different. The "third" argument was annotated as a direct dependent of the light verb (probably due to lower confidence of non-native speaker annotators in judging verb valency issues).

There is probably no chance of dealing with the dependencies in one of the two above stated ways only. The class of complex predicates in the data is wide and heterogeneous with respect to semantic and morphosyntactic qualities. Nevertheless, though resigning on the absolute consistency of the class, we may reach at least the consistency within the treatment of the individual light verbs throughout the corpus.

7.3. Conversive Verbs

A considerable number of unaligned arguments in the data is caused by the translator's choice of a verb in a conversive relation to the verb used in the original language. For some reason (e.g., frequency of the verbal lexical unit, topic-focus articulation etc.), the translator decides not to use the syntactically most similar lexical unit, but uses a conversive one (cf. also Sect. 7.1.2), thus causing the arguments to relocate in the deep syntactic structure, see Fig. 11.

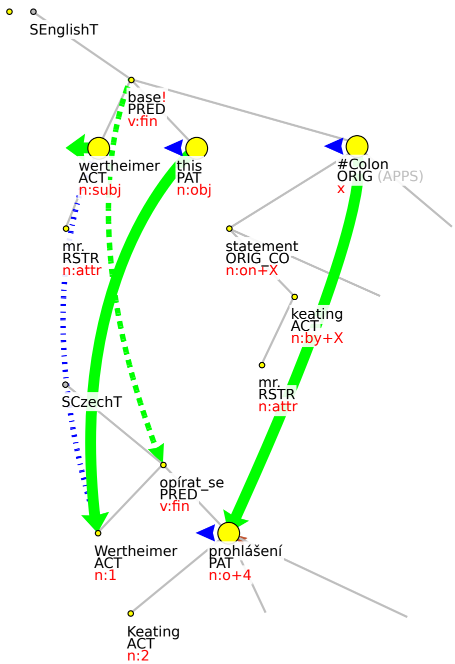


En: His election increases Ryder's board to 14 members.
Cz: Jeho zvolením se počet členů správní rady společnosti Ryder zvýšíl na 14.

Figure 11. Mismatch due to the use of conversive verbs

The relocation of arguments frequently goes together with backgrounding of one of the arguments, which then either disappears from the translation, or is transformed into an adjunct, or into a dependent argument embedded even lower in the structure.

The first actant (ACT) in the FGD approach is strongly underspecified. It is mostly delimited by its position in the tectogrammatic annotation. Its prevalent morphosyntactic realization is nominative case, but certain exceptions are recognized (verbs of feeling etc.). Also, the ACT position is subject to the process called "shifting of cognitive roles" (Panevová, 1974), cf. Sec. 2.1, i.e., other semantic roles can take the nominative case and the corresponding place in the structure in case there is no semantic agent in the structure. Thus we get semantically quite different elements (e.g., +anim vs. -anim) in the `ACT` position, even with formally identical verb instances (Fig. 12 and 13).

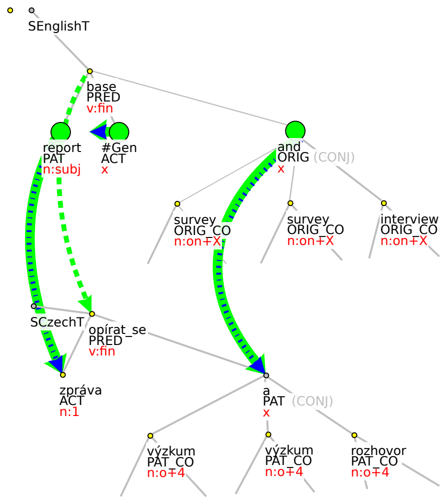


En: Mr. Wertheimer based this on a statement by Mr. Keating...
Cz: Wertheimer se opírá o prohlášení Keatinga...

Figure 12. Conflict due to the underspecification of the ACT position

This formal feature of the FGDVT gives rise to a number of conflicts in the parallel structures considering structures that undergo semantic de-agentization or (milder) de-concretization of the agent.

Here the question arises, whether such verb instances correspond to different meanings of the verb, or whether they correspond to a single meaning (represented by a single valency frame). It is often the case, that the Czech data tend to overgeneralize the valency frames through considering the different instances as realizations of a single deep syntactic valency frame, when there is no other modification intervening in the frame. Therefore, this approach chosen for the Czech annotation sometimes shows a conflict, as in Fig. 12 and 13.



En: The report was based on a telephone survey...
Cz: Zpráva se opírá o telefonický výzkum...

Figure 13. Original collect for the verbs base *and* opírat se

The valency structure for both instances of *base* (in Fig. 12 and 13) is identical, only in the first case, the verb is used in active voice, whereas in the second case, it is in passive voice. There are three semantic arguments in the structure. We will call them the Person that expresses an opinion, the Expressed Opinion and the Resource for the opinion. The Person bases the Expressed Opinion on the Resource. With the English verb, the Expressed Opinion always takes the `PAT` position and the Resource the `ORIG`in position in the valency structure. On the other hand, on the Czech side of the data, there is a conflict. In both Czech cases, there are seemingly only two arguments. In the first case, the Expressed Opinion is sort of backgrounded from the semantic structure. In the second case, on the other hand, the structure follows the passivized English structure in backgrounding the Person, the Expressed Opinion does not take the PAT position, but the ACT position in the structure, which is the cause of the conflict (for more details, see Šindlerová et al. (2015)).

The conflicts in annotation have a substantial reason – the ways in which English and Czech express backgrounding of the agent are multiple and they differ across the languages. Czech uses the *se*-morphemization often, in order to preserve the topic focus articulation (information) structure, whereas English does not have such a morpheme to work with, so it often uses simple passivization, or middle construction.

Moreover, the first valency position in Czech is often overgeneralized, allowing a multitude of semantically different arguments, which is, due to "economy of description", sometimes not reflected in the linguistic theory.

7.4. Head-dependent switch

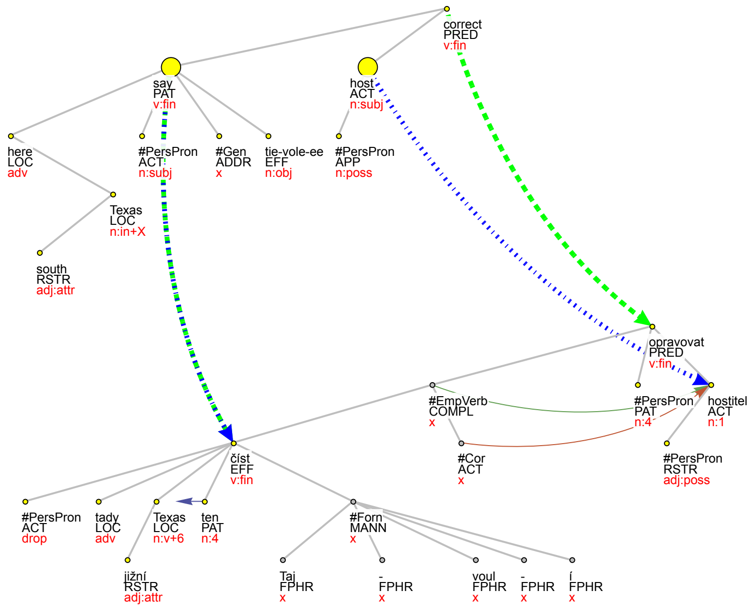
Due to some differences in annotation guidelines for the two languages, or due to translation issues, some slight semantic "switches" in alignments are allowed in order to map the arguments properly.

A frequent case of a head-dependent switch involves numerical expressions. For example, the English phrase *many economists* is annotated with *economist* as a head (labeled as argument) but in its Czech translation *řada ekonomů*, the word *řada* is, on the basis of its morphosyntactic behaviour, considered the head (labeled as valency argument), with *economist* in a dependent position. Numerical expressions overtaking the head position (with certain morphosyntactic consequences for the sentence) are called "container" expressions. With container expression of one side of translation, and modifying numeral on the other side, the alignment should be considered as encompassing a small subtree as opposed to a single node. Nevertheless, the annotators were asked to align head to head (i.e., align both direct daughters of the verb and arguments). In the above example, the word *economist* and *řada* are aligned instead of aligning the English head (*economist*) with the Czech dependent (*ekonom*) according to the very meaning of the lexical items, see Fig. 3 on page 30.

Another manifestation of the problem comes with the names of companies (e.g., *IBM*). Due to preservation of an appropriate inflection marking in the Czech translation, they are usually preceded with a generic name like *společnost* (*company*) in the Czech sentence, whereas they are used on their own in the English version of the sentence. In such cases, the alignment again is to be viewed as covering the whole subtree in Czech, and thus the nodes *IBM* and *společnost* are aligned.

7.5. Direct speech

According to the annotation guidelines, the annotation rules for direct speech in English (Cinková et al., 2006) and Czech (Mikulová et al., 2006a) on the tectogrammatical level are similar. Both languages add a new node representing the gerund (transgressive) of a verb of saying to the tectogrammatical annotation in cases where



En: “Here in south Texas we say Tie-vole-ee,” my host ... corrects .
Cz: ”Tady v jižním Texasu to čteme Taj-voul-í,” ... mě opravuje můj hostitel.

Figure 14. Direct speech alignment

the direct speech is adjacent to a verb which cannot be considered a verb reporting the direct speech (none of the arguments of the valency frame of the verb can be expressed by the direct speech). This newly added node is assigned a `t_lemma` substitute `#EmpVerb` and the functor `COMPL`. An example of a direct speech paraphrasable with a verb of saying: *Vtrhl do dveří* #EmpVerb.`COMPL`: *„Kdy bude.*`EFF` *večeře?“* (*He burst in at the door: “When will the dinner be ready?”*)

Due to the same instructions, mismatches were not expected in collecting direct speech utterances. Nevertheless, the annotation process reveals some discrepancies, as shown in Fig. 14, where the collected frame pair is as follows: `ACT`⟶`ACT PAT`⟶`---`, `---`⟶`PAT`.

The mismatch occurs due to a different practical annotation approach to direct speech in the individual languages, most notably, the English annotation often deviates from the common guidelines. While in Czech the use of `#EmpVerb` and the functor `COMPL` is common, in English the addition of the `#EmpVerb` node is rarely done.

In case of such a discrepancy in the data, based on the presence of a `COMPL` node on just one side of the translation, the annotator is asked neither to align the direct argument of the other side to the COMPL node, nor to its lexical counterpart, but rather to collect the zero alignment (alignment to no specific node in the structure, see Sec. 6.2.2). Such structures are left for future treatment within possible tectogrammatical annotation revisions.

8. Use and future work

The CzEngVallex has been planned as a resource to be used both for the purposes of possibly revising theoretical linguistic accounts of verbal valency from a crosslinguistic perspective, and for an innovative use in various NLP tasks.

In both of these areas, the CzEngVallex has proved to be a valid resource. Our publications Šindlerová et al. (2013); Urešová et al. (2013); Šindlerová et al. (2014); Urešová et al. (2014a, 2015); Šindlerová et al. (2015); Urešová et al. (2015) show some interesting and important results concerning verbal valency from the Czech-English comparison perspective, while Dušek et al. (2014, 2015) shows that the inclusion of the CzEngVallex bilingual mapping feature into a word sense disambiguation task significantly improves the performance of the system. Our findings are also very useful when comparing different formal representations of meaning, see Xue et al. (2014); Urešová et al. (2014b); Oepen et al. (2015).

As for future work, a more detailed comparative description of the argument structure of translation equivalents found in the data would be needed. The attention should be paid especially to verb–non-verb or non-verb–verb pairs which were not included in the first version of CzEngVallex. And, of course, there exist many other manifestations of the above mentioned phenomena: functor mismatches, conflicts in data, zero alignments, which deserve our future attention and which might - on top of their better understanding from the linguistic point of view - lead to changes in the structure and content of the underlying valency lexicons towards a more universal valency description with less differences across languages. The results could also influence translation studies and the practice of translation, as well as deep methods in the area of natural language processing.

We also plan to create (manually but with substantial computational support) a class-based "superlexicon" over the CzEngVallex, grouping together synonyms or at least related sense pairs.

Acknowledgements

We would like to thank the reviewers for their insightful comments, that help improve the paper. This work described herein has been supported by the grant GP13-03351P of the Grant Agency of the Czech Republic and it is using language resources hosted by the LINDAT/CLARIN Research Infrastructure, project LM2010013 funded by the MEYS of the Czech Republic.

Bibliography

Bojar, Ondřej and Jana Šindlerová. Building a Bilingual ValLex Using Treebank Token Alignment: First Observations. In *Proceedings of the 7th International Conference on Language Resources and Evaluation (LREC 2010)*, pages 304–309, Valletta, Malta, 2010. ELRA.

Cinková, Silvie. From PropBank to EngValLex: Adapting the PropBank-Lexicon to the Valency Theory of the Functional Generative Description. In *Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC 2006)*, pages 2170–2175, Genova, Italy, 2006. ELRA, ELRA. ISBN 2-9517408-2-4.

Cinková, Silvie, Jan Hajič, Marie Mikulová, Lucie Mladová, Anja Nedolužko, Petr Pajas, Jarmila Panevová, Jiří Semecký, Jana Šindlerová, Josef Toman, Zdeňka Urešová, and Zdeněk Žabokrtský. Annotation of English on the tectogrammatical level. Technical Report 35, UFAL MFF UK, 2006.

Dušek, Ondřej, Jan Hajič, and Zdeňka Urešová. Verbal Valency Frame Detection and Selection in Czech and English. In *The 2nd Workshop on EVENTS: Definition, Detection, Coreference, and Representation*, pages 6–11, Stroudsburg, PA, USA, 2014. Association for Computational Linguistics. ISBN 978-1-941643-14-3.

Dušek, Ondřej, Eva Fučíková, Jan Hajič, Martin Popel, Jana Šindlerová, and Zdeňka Urešová. Using Parallel Texts and Lexicons for Verbal Word Sense Disambiguation. In Hajičová, Eva and Joakim Nivre, editors, *Proceedings of the Third International Conference on Dependency Linguistics (Depling 2015)*, pages 82–90, Uppsala, Sweden, 2015. Uppsala University, Uppsala University. ISBN 978-91-637-8965-6.

Hajič, Jan, Eva Hajičová, Jarmila Panevová, Petr Sgall, Silvie Cinková, Eva Fučíková, Marie Mikulová, Petr Pajas, Jan Popelka, Jiří Semecký, Jana Šindlerová, Jan Štěpánek, Josef Toman, Zdeňka Urešová, and Zdeněk Žabokrtský. Prague Czech-English Dependency Treebank 2.0, 2011.

Kingsbury, P. and M. Palmer. From Treebank to Propbank. In *Proceedings of the 3rd International Conference on Language Resources and Evaluation (LREC-2002)*, pages 1989–1993. Citeseer, 2002.

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., 2006a.

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., 2006b.

Mindt, Dieter. Finite vs. Non-Finite Verb Phrases in English. In *Form, Function and Variation in English*, pages 343–352, Frankfurt am Main, 1999. Peter Lang GmbH. ISBN 978-3-631-33081-4.

Oepen, Stephan, Marco Kuhlmann, Yusuke Miyao, Daniel Zeman, Silvie Cinková, Dan Flickinger, Jan Hajič, and Zdeňka Urešová. SemEval 2015 Task 18: Broad-Coverage Semantic Dependency Parsing. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 915–926, Denver, Colorado, June 2015. Association for Computational Linguistics. ISBN 978-1-941643-40-2. URL `http://aclweb.org/anthology/S15-2153`.

Pajas, Petr and Peter Fabian. TrEd 2.0 - newly refactored tree editor. `http://ufal.mff.cuni.cz/tred`, 2011.

Palmer, F. R. *The English verb / F. R. Palmer*. Longman London, 2d ed. edition, 1974. ISBN 058252458.

Palmer, Martha, Dan Gildea, and Paul Kingsbury. The proposition bank: An annotated corpus of semantic roles. *Computational Linguistics*, 31(1):71–106, 2005.

Panevová, Jarmila. On verbal frames in Functional generative description I. *Prague Bulletin of Mathematical Linguistics*, 22:3–40, 1974.

Panevová, J. On verbal Frames in Functional Generative Description. *Prague Bulletin of Mathematical Linguistics*, 22:3–40, 1974.

Panevová, Jarmila. More Remarks on Control. *Prague Linguistic Circle Papers*, 2(1):101–120, 1996.

Popel, Martin and Zdeněk Žabokrtský. TectoMT: modular NLP framework. *Advances in Natural Language Processing*, pages 293–304, 2010.

Przepiórkowski, Adam and Alexandr Rosen. Czech and Polish Raising/Control with or without Structure Sharing. *Research in Language*, 3:33–66, 2005.

Šindlerová, Jana and Ondřej Bojar. Towards English-Czech Parallel Valency Lexicon via Treebank Examples. In *Proceedings of 8th Treebanks and Linguistic Theories Workshop (TLT)*, pages 185–195, Milano, Italy, 2009.

Šindlerová, Jana, Zdeňka Urešová, and Eva Fučíková. Verb Valency and Argument Noncorrespondence in a Bilingual Treebank. In Gajdošová, Katarína and Adriána Žáková, editors, *Proceedings of the Seventh International Conference Slovko 2013; Natural Language Processing, Corpus Linguistics, E-learning*, pages 100–108, Lüdenscheid, Germany, 2013. Slovak National Corpus, Ľ. Štúr Institute of Linguistics, Slovak Academy of Sciences, RAM-Verlag. ISBN 978-3-942303-18-7.

Šindlerová, Jana, Zdeňka Urešová, and Eva Fučíková. Resources in Conflict: A Bilingual Valency Lexicon vs. a Bilingual Treebank vs. a Linguistic Theory. In Calzolari, Nicoletta, Khalid Choukri, Thierry Declerck, Hrafn Loftsson, Bente Maegaard, and Joseph Mariani, editors, *Proceedings of the 9th International Conference on Language Resources and Evaluation (LREC 2014)*, pages 2490–2494, Reykjavík, Iceland, 2014. European Language Resources Association. ISBN 978-2-9517408-8-4.

Šindlerová, Jana, Eva Fučíková, and Zdeňka Urešová. Zero Alignment of Verb Arguments in a Parallel Treebank. In Hajičová, Eva and Joakim Nivre, editors, *Proceedings of the Third International Conference on Dependency Linguistics (Depling 2015)*, pages 330–339, Uppsala, Sweden, 2015. Uppsala University.

Urešová, Zdeňka. *Valence sloves v Pražském závislostním korpusu*. Studies in Computational and Theoretical Linguistics. Ústav formální a aplikované lingvistiky, Praha, Czechia, 2011a.

Urešová, Zdeňka. *Valenční slovník Pražského závislostního korpusu (PDT-Vallex)*. Studies in Computational and Theoretical Linguistics. Ústav formální a aplikované lingvistiky, Praha, Czechia, 2011b.

Urešová, Zdeňka, Eva Fučíková, Jan Hajič, and Jana Šindlerová. An Analysis of Annotation of Verb-Noun Idiomatic Combinations in a Parallel Dependency Corpus. In *The 9th Workshop on Multiword Expressions (MWE 2013)*, pages 58–63, Atlanta, Georgia, USA, 2013. Association for Computational Linguistics. ISBN 978-1-937284-47-3.

Urešová, Zdeňka, Eva Fučíková, Jan Hajič, and Jana Šindlerová. Verb-Noun Idiomatic Combinations in a Czech-English Dependency Corpus. In *PARSEME 2nd general meeting*, Athens, Greece, 2014a. Institute for Language and Speech Processing of the Athena Research Center.

Urešová, Zdeňka, Jan Hajič, and Ondřej Bojar. Comparing Czech and English AMRs. In *Proceedings of Workshop on Lexical and Grammatical Resources for Language Processing (LG-LP 2014, at Coling 2014)*, pages 55–64, Dublin, Ireland, 2014b. Dublin City University, Association for Computational Linguistics and Dublin City University. ISBN 978-1-873769-44-7.

Urešová, Zdeňka, Ondřej Dušek, Eva Fučíková, Jan Hajič, and Jana Šindlerová. Bilingual English-Czech Valency Lexicon Linked to a Parallel Corpus. In *Proceedings of the The 9th Linguistic Annotation Workshop (LAW IX 2015)*, Stroudsburg, PA, USA, 2015. Association for Computational Linguistics.

Urešová, Zdeňka, Eva Fučíková, Jan Hajič, and Jana Šindlerová. CzEngVallex, 2015. URL `http://hdl.handle.net/11234/1-1512`. LINDAT/CLARIN digital library at Institute of Formal and Applied Linguistics, Charles University in Prague.

Urešová, Zdeňka, Eva Fučíková, and Jana Šindlerová. CzEngVallex: Mapping Valency between Languages. Technical Report TR-2015-58, ÚFAL MFF UK, 2015.

Xue, Nianwen, Ondřej Bojar, Jan Hajič, Martha Palmer, Zdeňka Urešová, and Xiuhong Zhang. Not an Interlingua, But Close: Comparison of English AMRs to Chinese and Czech. In Calzolari, Nicoletta, Khalid Choukri, Thierry Declerck, Hrafn Loftsson, Bente Maegaard, and Joseph Mariani, editors, *Proceedings of the 9th International Conference on Language Resources and Evaluation (LREC 2014)*, pages 1765–1772, Reykjavík, Iceland, 2014. European Language Resources Association. ISBN 978-2-9517408-8-4.

Žabokrtský, Zdeněk. Treex – an open-source framework for natural language processing. In Lopatková, Markéta, editor, *Information Technologies – Applications and Theory*, volume 788, pages 7–14, Košice, Slovakia, 2011. Univerzita Pavla Jozefa Šafárika v Košiciach. ISBN 978-80-89557-02-8.

Address for correspondence:
Zdeňka Urešová
`uresova@ufal.mff.cuni.cz`
Institute of Formal and Applied Linguistics
Faculty of Mathematics and Physics, Charles University in Prague
Malostranské náměstí 25, 118 00 Praha 1, Czech Republic

PBML

The Prague Bulletin of Mathematical Linguistics

NUMBER 105 APRIL 2016 51–61

CloudLM: a Cloud-based Language Model for Machine Translation

Jorge Ferrández-Tordera[a], Sergio Ortiz-Rojas[a], Antonio Toral[b]

[a] Prompsit Language Engineering S.L., Elche, Spain
[b] ADAPT Centre, School of Computing, Dublin City University, Ireland

Abstract

Language models (LMs) are an essential element in statistical approaches to natural language processing for tasks such as speech recognition and machine translation (MT). The advent of big data leads to the availability of massive amounts of data to build LMs, and in fact, for the most prominent languages, using current techniques and hardware, it is not feasible to train LMs with all the data available nowadays. At the same time, it has been shown that the more data is used for a LM the better the performance, e.g. for MT, without any indication yet of reaching a plateau. This paper presents CloudLM, an open-source cloud-based LM intended for MT, which allows to query distributed LMs. CloudLM relies on Apache Solr and provides the functionality of state-of-the-art language modelling (it builds upon KenLM), while allowing to query massive LMs (as the use of local memory is drastically reduced), at the expense of slower decoding speed.

1. Introduction

Language models (LMs) are an essential element in statistical approaches to natural language processing for tasks such as speech recognition and machine translation (MT). The advent of big data leads to the availability of massive amounts of monolingual data, which could be used to build LMs. In fact, for the most prominent languages, using current techniques and hardware, it is not feasible to train LMs with all the data available nowadays. At the same time, it has been shown that the more data is used for a LM the better the performance, e.g. for MT, without any indication yet of reaching a plateau (Brants et al., 2007).

 Corresponding author: jferrandez@prompsit.com
Cite as: Jorge Ferrández-Tordera, Sergio Ortiz-Rojas, Antonio Toral. CloudLM: a Cloud-based Language Model for Machine Translation. The Prague Bulletin of Mathematical Linguistics No. 105, 2016, pp. 51–61.
doi: 10.1515/pralin-2016-0002.

Our aim in this paper is to build a cloud-based LM architecture, which would allow to query distributed LMs on massive amounts of data. Our architecture is called CloudLM, it is open-source, it is integrated in the Moses MT toolkit[1] and is based on Apache Solr.[2]

The rest of the paper is organised as follows. In Section 2 we provide an overview of the state-of-the-art in huge LMs. Next, Section 3 details our architecture. This is followed by a step-by-step guide to CloudLM in Section 4 and its evaluation in terms of efficiency in Section 5. Finally, we conclude and outline avenues of future work in Section 6.

2. Background

Brants et al. (2007) presented a distributed architecture with the aim of being able to use big data to train LMs. They trained a LM on 2 trillion tokens of text with simplified smoothing, resulting in a 5-gram language model size of 300 billion n-grams. The infrastructure used in their experiment involved 1,500 machines and took 1 day to build the LM. It is worth mentioning that the infrastructure is scalable, so one could use more machines to train LMs on larger amounts of data and/or LMs of higher n-gram orders.

Talbot and Osborne (2007) investigate the use of the Bloom filter, a randomised data structure, to build n-gram-based LMs. Compared to conventional n-gram-based LMs, this approach results in considerably smaller LMs, at the expense, however, of slower decoding. This approach has been implemented in RandLM,[3] which supports distributed LMs.

More recent work explores the training of huge LMs on single machines (Heafield et al., 2013). The authors build a LM on 126 billion tokens, with the training taking 123 GB of RAM, 2.8 days wall time, and 5.4 CPU days. A machine with 1 TB RAM was required to tune an MT system that uses this LM (Durrani et al., 2014).

Memory mapping has been used to reduce the amount of memory needed by huge LMs, at the expense of slower MT decoding speed (Federico and Cettolo, 2007). In the experiments conducted in that work, memory mapping led to decrease the amount of memory required in half at the cost of 44% slower decoding.

The most related previous work to ours is Brants et al. (2007). There are two main differences though: (i) that work relied on a simplified smoothing technique to enhance efficiency while CloudLM uses state-of-the-art smoothing techniques and (ii) our work is open-source and is integrated in the Moses statistical MT toolkit.

[1] `https://github.com/jferrandez/mosesdecoder/tree/cache-cloudlm`

[2] `http://lucene.apache.org/solr/`

[3] `http://randlm.sourceforge.net/`

3. Architecture

This section describes the architecture of CloudLM. First we cover the representation of LMs in Apache Solr (Section 3.1). Then we detail the implementation of CloudLM in the Moses toolkit (Section 3.2). Finally, we describe two efficiency enhancements that have been added to CloudLM, a cache and queries (Section 3.3).

3.1. LMs in Solr

In order to have LMs in Solr, we need to represent in a Solr schema the fields of an ARPA LM entry,[4] namely: (i) the n-gram, (ii) its probability, (iii) its back-off weight and (iv) its order. We define these fields in a Solr schema as shown in Figure 1.

```
<field name="ngram"   type="string" indexed="true"  stored="true"/>
<field name="prob"    type="float"  indexed="false" stored="true"/>
<field name="backoff" type="float"  indexed="false" stored="true"/>
<field name="order"   type="int"    indexed="true"  stored="true"/>
```

Figure 1. ARPA fields in Solr schema

The fields `ngram` and `order` are indexed (`indexed="true"`) as those are the ones we use to query the LM. All the fields are stored (`stored="true"`) meaning that they can be returned by queries.

3.2. Cloud-based LM in Moses

CloudLM is implemented as a new module in Moses that builds upon KenLM (Heafield, 2011). In short, we adapt KenLM's functions that query n-grams on ARPA or binary files so that they query our cloud-based model instead and we remove any other files that are not required for querying LMs (e.g. build and binarise LMs, trie models, quantise, etc.). As a result, given a query and a LM, the output produced by CloudLM and KenLM are exactly the same.

CloudLM provides two main advantages as a result of its distributed nature: (i) there is no loading time and (ii) memory requirements in the machine where decoding takes place are considerably reduced. That said, there is an important disadvantage, in that its use results in slower MT decoding speed.

[4] `http://www1.icsi.berkeley.edu/Speech/docs/HTKBook3.2/node213_mn.html`

3.3. Efficiency Enhancements

In order to mitigate the main disadvantage of CloudLM (its lower querying speed), we implement two efficiency enhancements, a cache (Section 3.3.1) and a block query (Section 3.3.2).

3.3.1. Cache

Caches are known to be useful in any network dependent process (thus subject to high latency) that requests repeated queries. In CloudLM we implement a cache in order to avoid several queries requesting the probability for the same n-gram.

Intuitively, the advantage of the cache is that it should save time due to network latency. However, the data stored in the cache structure should lead to higher requirements of memory.

Our selected cache strategy is least recently used (LRU), in which the least recently used items are discarded first. In the way that Moses queries the LM, LRU guarantees that the most recently requested n-grams will be found in the cache.

3.3.2. Block Query

As we adapt KenLM querying functions only with respect to the repository where the LM is stored (from local files to Solr), queries are still submitted individually for each n-gram. For example, given the 2-gram "we are", three queries would be submitted to the LM: one for the bi-gram "we are" and two for the 1-grams "we" and "are". Our first approach for using the cache is to store the probability returned for this 2-gram.

In order to minimise the amount of queries sent to Solr (and saving network latency), we implement a block n-grams query. When the LM receives a phrase, we extract all its possible n-grams and prepare a query that contains them all. For instance, for the previous example, we prepare a query with the 2-gram "we are" and the 1-grams "we" and "are". In this way we can retrieve the three probabilities with one single query.

4. Step-by-Step

In this section we provide a step-by-step guide to use CloudLM in Moses. We assume we have a Moses system (Koehn et al., 2007) trained (translation and reordering models), e.g. according to Moses baseline guide,[5] an LM ready in ARPA format, e.g. trained with KenLM, and an installation of Apache Solr. The steps are as follows:

1. Configure Solr.

[5] `http://www.statmt.org/moses/?n=Moses.Baseline`

The LM can be placed in the local machine, in a remote one, or be distributed across a number of machines. We cover each of these cases in the following:

- Local machine. While the main advantage of using Solr relies in its distributed nature, we can still use it locally, where Solr's advantage will be its lower use of memory (as the LM is not loaded completely in RAM).
- Remote machine. In this case Solr is used from one remote machine. This can be useful when the local machine does not have enough resources for the LM but we have access to a remote machine with enough resources.
- Distributed architecture. Solr allows to have the LM distributed across a number of machines.[6] This can be useful when we have access to a number of remote machines and we have to deal with a huge LM that does not fit on any of those machines alone.

2. Upload LM to Solr. This is done with a script included with CloudLM that reads an ARPA file, converts it to Solr Schema (cf. Section 3.1) and uploads it to a Solr installation.

```
python add-language-model-from-arpa.py \
       http://localhost:8983/solr lm.arpa
```

3. Include CloudLM in Moses' configuration (ini file). The format is very similar to that of KenLM, the only three differences being that (i) the LM type is `CLOUDLM` (instead of `KENLM`), that (ii) the LM is indicated by means of a URL (instead of a local path) and that (iii) there is a binary variable to indicate whether or not to use a cache (cf. Section 3.3.1).

```
CLOUDLM name=LM0 factor=0 order=4 \
        num-features=1 cache=0 url=localhost:8983
```

From this point onward, we can proceed with tuning and decoding with Moses as one would normally do.

5. Experiments

In this section we conduct a set of experiments in order to measure efficiency of CloudLM in terms of computational resources (real time and memory) in the task of statistical MT. First, we detail the experimental setting (Section 5.1) and then we present the results for three experiments (Section 5.2) where we measure (i) the effect of the efficiency enhancements on top of CloudLM, (ii) the effect of network latency and finally we (iii) compare the efficiency of CloudLM to that of a state-of-the-art *local* LM, KenLM.

---

[6] https://cwiki.apache.org/confluence/display/solr/SolrCloud

5.1. Experimental Setting

The MT systems built for our experiments fall into the statistical phrase-based paradigm and they are built with Moses version 3 following the baseline system guideline.[7] All the systems are trained on the Europarl v7 (Koehn, 2005) parallel corpus for the language direction English-to-Spanish. All the LMs are built on the Spanish side of that parallel corpus. [8] We use these MT systems to decode subsets (1, 10 and 100 sentences) of the test set from WMT13.[9]

We use both a local and a remote machine in our experiments.[10] The local machine has a 8-core i7-3632QM CPU at 2.20GHz, 16GB RAM and a SATA 3.0 500GB hard drive. The remote machine has 4-core Q8200 CPU at 2.33GHz, 4GB RAM and a SATA 3.0 1TB hard drive.

5.2. Results

In all the experiments below we measure the peak of memory used and real time required to translate the first 1, 10 and 100 sentences of the testset with the different systems evaluated.

5.2.1. Effect of Enhancement Additions

In this experiment we measure the effect of the efficiency enhancements that have been added to CloudLM, namely the cache (cf. Section 3.3.1) and block queries (cf. Section 3.3.2). We build three systems where the LMs are built with CloudLM using different settings: stock (reported in results below as S), with cache (WC) and with both cache and block queries (WCBQ). All the LMs are stored locally.

Figure 2 reports the real time and Moses' memory peak required by each system to decode the first 1, 10 and 100 sentences from the test set. The use of cache, as expected, results in a notable reduction in time but also increases memory usage. For 1 sentence, using cache reduces the time by around 70% and memory used augments by 20%. These figures increase with the number of sentences decoded; with 100 sentences the use of cache reduces the time required by 89% while memory used increments by 195%. On its turn, the use of block queries (WCBQ) provides a slight time advantage when decoding 1 sentence (9% faster), but it is slower for 10 and 100 sentences. We are currently investigating the causes for this.

Table 1 provides further details regarding the use of cache and block queries. It shows the total number of requests submitted to Solr (column # requests), the number

[7] `http://www.statmt.org/moses/?n=Moses.Baseline`

[8] The existing model indexed in Solr takes 1.95 GB. The original binarized ARPA file amounts to 830 MB.

[9] `http://www.statmt.org/wmt13/translation-task.html`

[10] Before each run the machines were rebooted to ensure data from the previous run is not leveraged from the disk cache.

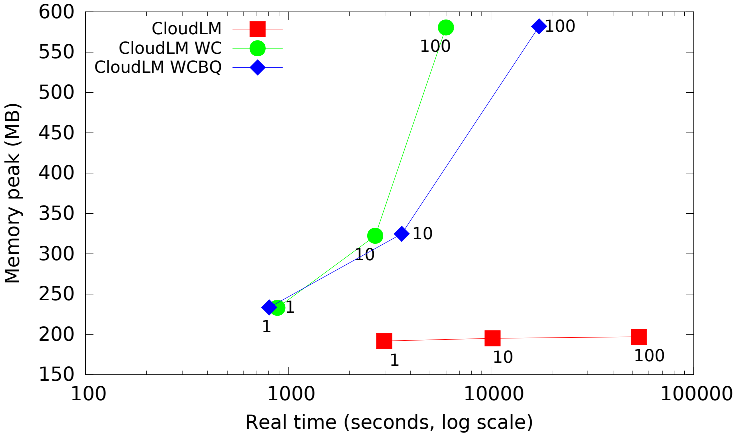


Figure 2. Effect of enhancement additions

of queries that are stored in the cache (# insertions), the number of lookups in the cache (# lookups) and the percentage of successful lookups (% found), i.e. the cache contains the query requested and hence the query is not submitted to Solr. The use of the cache reduces drastically the number of queries sent to Solr, even when translating just one sentence this number is reduced by 85%.The use of block queries reduces even more the amount of queries sent, as the percentage of queries found in the cache is even higher (e.g. 99.8% for 1 sentence).

# sents.	System	# requests	# inserts	# lookups	% found
1	S	1,779,225	0	0	0
1	WC	264,851	264,851	1,779,225	85.11
1	WCBQ	206,160	264,851	1,779,225	99.80
10	S	7,067,343	0	0	0
10	WC	822,627	822,627	7,067,343	88.36
10	WCBQ	929,020	822,627	7,067,343	98.21
100	S	22,417,996	0	0	0
100	WC	2,417,593	2,417,593	22,417,996	89.21
100	WCBQ	4,493,867	2,417,593	22,417,996	94.45

Table 1. Effects of the use of cache and block queries with CloudLM.

5.2.2. Effect of Network Latency

In this experiment we measure the effect of network latency. Clearly, an advantage of CloudLM relies in the fact that it allows us to use LMs placed in remote machines. Accessing them, though, penalises efficiency as each query is subject to network latency.

We use two systems, both using CloudLM with cache. One of the systems accesses the LM locally while the other accesses it from a remote machine in the local network.

Figure 3 reports the real time and memory peak required by each system to decode different amounts of sentences. Network latency affects efficiency quite drastically; accessing the LM from a remote machine results in decoding speed an order of magnitude slower. We measured the average latency in the local and remote machines used in this experiment. The figures were 0.04 milliseconds for the local machine and 0.277 for the remote one.

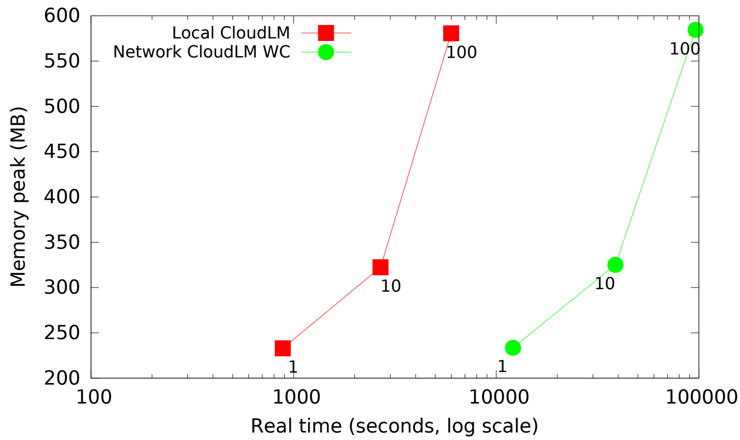


Figure 3. Effect of network latency

5.2.3. Comparison to a *local* LM

Finally, we compare, in terms of efficiency, CloudLM to a state-of-the-art *local* LM, KenLM. We have three systems, one with CloudLM (with cache), and two with KenLM (with and without loading on demand, reported in the results as lazy KenLM and KenLM respectively). All LMs are stored locally.

Figure 4 shows the results. CloudLM reduces notably the amount of memory required at the expense of the decoding speed becoming between one and two orders of magnitude higher. For one sentence, CloudLM is 70 times slower (240 compared to KenLM on demand) and reduces the amount of memory required by 77% (65% compared to on demand). As we add more sentences the differences on both speed and memory shrink, with CloudLM being 33 times slower (68 compared to the on demand version of KenLM) and reducing the amount of memory by 46% (45% compared to KenLM on demand) for 100 sentences.

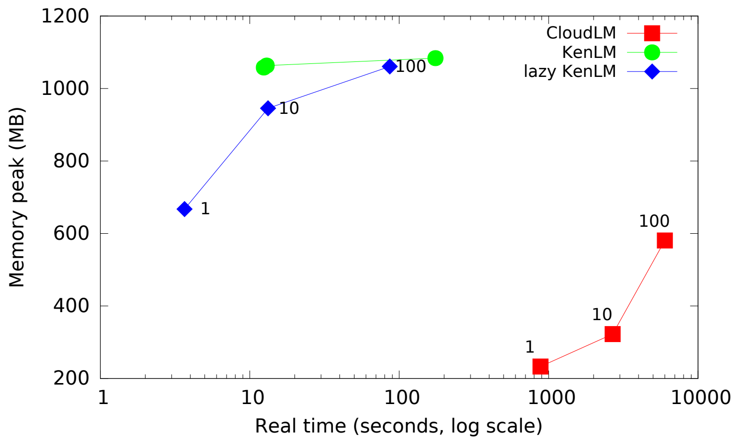


Figure 4. Comparison of CloudLM to a local LM

6. Conclusions and Future Work

This paper has presented CloudLM, an open-source cloud-based LM that allows to build distributed LMs for their use in e.g. MT. CloudLM is based on Apache Solr and KenLM, providing the functionality of the latter in a distributed environment. The focus of our work so far has been on providing a stable and robust implementation that can be extended upon to make it more efficient.

The current implementation uses a simple cache model (LRU) and can send joint queries in order to diminish the efficiency penalty posed by network latency. We have evaluated CloudLM in terms of efficiency to measure the effect of the efficiency additions, the effect of latency and finally to compare its use of resources compared to a state-of-the-art *local* LM.

We envisage two main lines of future work. First, development work to enhance efficiency. We have several ideas in this regard, such as keeping the connection alive between Moses and Solr (so that a new query does not need to re-open the connection) and using more advance cache strategies. The efficiency bottleneck in a synchronous distributed architecture like ours has to do with the network latency. Hence, we propose to have an asynchronous connection instead, so that Moses does not need to wait for each response from Solr. This, however, is far from straightforward as it would entail deeper modifications to the MT decoder.

Our second line of future work has to do with the evaluation of CloudLM for huge LMs. The evaluation in the current paper can be considered as proof-of-concept, as we have dealt with a rather small LM (around 2 million sentence pairs).

Finally, we would like to compare CloudLM to other approaches that use distributed LMs in Moses (Federmann, 2007; Talbot and Osborne, 2007). Such an evaluation would not be purely efficiency-based (e.g. decoding time, memory used) but also would take into account the final translation quality achieved as some of these approaches use different modelling techniques (e.g. Bloom filter in RandLM).

Acknowledgments

The research leading to these results has received funding from the European Union Seventh Framework Programme FP7/2007-2013 under grant agreement PIAP-GA-2012-324414 (Abu-MaTran).

Bibliography

Brants, Thorsten, Ashok C. Popat, Peng Xu, Franz Josef Och, and Jeffrey Dean. Large Language Models in Machine Translation. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 858–867, 2007. URL `http://www.aclweb.org/anthology/D/D07/D07-1090`.

Durrani, Nadir, Barry Haddow, Philipp Koehn, and Kenneth Heafield. Edinburgh's Phrase-based Machine Translation Systems for WMT-14. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 97–104, Baltimore, Maryland, USA, June 2014. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/W/W14/W14-3309`.

Federico, Marcello and Mauro Cettolo. Efficient Handling of N-gram Language Models for Statistical Machine Translation. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 88–95, Prague, Czech Republic, June 2007. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/W/W07/W07-0212`.

Federmann, Christian. Very Large Language Models for Machine Translation. Master's thesis, Saarland University, Saarbrücken, Germany, July 2007. URL `http://www.cfedermann.de/pdf/diploma-thesis.pdf`.

Heafield, Kenneth. KenLM: Faster and Smaller Language Model Queries. In *Proceedings of the EMNLP 2011 Sixth Workshop on Statistical Machine Translation*, pages 187–197, Edinburgh, Scotland, United Kingdom, July 2011. URL `http://kheafield.com/professional/avenue/kenlm.pdf`.

Heafield, Kenneth, Ivan Pouzyrevsky, Jonathan H. Clark, and Philipp Koehn. Scalable Modified Kneser-Ney Language Model Estimation. In *Proceedings of the 51st Annual Meeting of the*

Association for Computational Linguistics, pages 690–696, Sofia, Bulgaria, August 2013. URL http://kheafield.com/professional/edinburgh/estimate_paper.pdf.

Koehn, Philipp. Europarl: A parallel corpus for statistical machine translation. In *MT summit*, volume 5, pages 79–86. Citeseer, 2005.

Koehn, Philipp, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. Moses: Open Source Toolkit for Statistical Machine Translation. In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, ACL '07, pages 177–180, Stroudsburg, PA, USA, 2007. Association for Computational Linguistics. URL http://dl.acm.org/citation.cfm?id=1557769.1557821.

Talbot, David and Miles Osborne. Smoothed Bloom Filter Language Models: Tera-Scale LMs on the Cheap. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 468–476, 2007. URL http://www.aclweb.org/anthology/D/D07/D07-1049.

Address for correspondence:
Jorge Ferrández-Tordera
jferrandez@prompsit.com
Avenida Universidad s/n Edificio Quorum III - Prompsit,
Elche, Alicante, Spain. 03202

PBML

The Prague Bulletin of Mathematical Linguistics

NUMBER 105 APRIL 2016 63–76

An Algorithm for Morphological Segmentation of Esperanto Words

Theresa Guinard

New Mexico Institute of Mining and Technology

Abstract

Morphological analysis (finding the component morphemes of a word and tagging morphemes with part-of-speech information) is a useful preprocessing step in many natural language processing applications, especially for synthetic languages. Compound words from the constructed language Esperanto are formed by straightforward agglutination, but for many words, there is more than one possible sequence of component morphemes. However, one segmentation is usually more semantically probable than the others. This paper presents a modified n-gram Markov model that finds the most probable segmentation of any Esperanto word, where the model's states represent morpheme part-of-speech and semantic classes. The overall segmentation accuracy was over 98% for a set of presegmented dictionary words.

1. Introduction

Esperanto, a planned language developed in 1887, is purely agglutinative; compound words are formed by juxtaposing morphemes, where the spelling and pronunciation of the morphemes do not change during this process. The official rules for word formation are permissive, but in practice, producing an understandable compound word relies on complex semantic relationships between the morphemes.

Sometimes, an Esperanto word is morphologically ambiguous: there is more than one grammatically legal sequence of component morphemes. For example, the word "katokulo" can be segmented as "kat'okul'o", meaning "cat eye", as "kat'o'kul'o", meaning "cat-like gnat", or as "kat'ok'ul'o", which is grammatically permissible (by official rules), but has no discernible meaning. Usually, one segmentation is more semantically probable than the others.

 Corresponding author: tguinard@gmail.com
Cite as: Theresa Guinard. An Algorithm for Morphological Segmentation of Esperanto Words. The Prague Bulletin of Mathematical Linguistics No. 105, 2016, pp. 63–76. doi: 10.1515/pralin-2016-0003.

This study confronts the problem of morphological analysis: segmenting a word into component morphemes, and tagging the morphemes with part-of-speech information. This study takes a supervised approach, so I assume that a lexicon of tagged morphemes is given. Because the process for forming compound words is purely agglutinative, one can easily find the set of all possible segmentations for a given Esperanto word, but the main challenge is disambiguation.

Morphological analysis can potentially benefit a wide range of natural language processing applications, as individual word structures and meanings become easier to systematically interpret. For highly agglutinative or highly inflectional language, this is especially useful. In particular, for such languages, morphological analysis has been successfully applied to spell checking algorithms (Agirre et al., 1992) and (Solak and Oflazer, 1992), and machine translation (Lee, 2004) and (Goldwater and McClosky, 2005).

2. Overview of Esperanto Morphology

Esperanto morphemes can be categorized into four general categories: word endings, roots, affixes, and standalone words.

Word endings mark the part of speech of most words as a noun, verb, adjective, or adverb. Word endings also incorporate inflectional information. The morpheme "j" indicates whether a noun or adjective is plural; the word ending for a noun is "o", but the word ending for a plural noun is "oj". The morpheme "n" can be added to a noun, adjective, or adverb ending to mark the accusative case; "on" would signify an accusative noun, and "ojn" would signify a plural accusative noun. The accusative marker can also be appended to pronouns, and the plural and accusative markers can be appended to some correlatives. There are exactly six word endings for verbs, which indicate different tenses and moods: "i", "os", "as", "is", "u", and "us" respectively correspond to the infinitive, future tense, present tense, past tense, imperative, and conditional forms of the verb.

Roots make up the majority of Esperanto morphemes. A root has no definite part of speech, so in principle, any root can be combined with any word ending. For example, the root "pluv" is often used as a noun: "pluvo" ("rain"). However, "pluvi" (verb; "to rain"), "pluva" (adjective; "rain-like"), and "pluve" (adverb; "like rain") are all permissible Esperanto words. Although any word ending can be used, Kalocsay and Waringhien (1985) proposed that each root has an inherent part of speech. Currently, the official morpheme list provided by Akademio de Esperanto (2008) implements this idea, listing each root with the most frequent word ending.

Affixes can legally function in the same way as roots, but are usually prepended or appended to roots. For example, the prefix "mal" ("opposite") negates the meaning of the word it prepends: "bona" ("good") becomes "malbona" ("bad"), but "mal" can also function as an ordinary root: "malo" (noun; "opposite"). Similarly the suffix "an" ("member") usually modifies a root: "klubo" ("club") becomes "klubano" ("club member"), but it can also form the word "ano" (noun; "member").

There is a notable class of suffixes, which are not used as roots in practice, but form participles to create compound tenses. One such suffix is "it", which can be appended to the verb root "skrib" ("to write") to form the phrase "estas skribita" ("has been written"). The suffixes in this class refer to different tenses ("has been written" vs. "is being written") and may refer to either the subject or object of the verb ("has been written" vs. "has been writing").

Standalone words are commonly-used words, including numbers, prepositions, pronouns, articles, exclamations, correlatives, and some adverbs. Correlatives are a class of function words including interrogatives ("what", "which"), demonstratives ("somehow", "somebody"), universals ("always", "everything"), and negatives ("nothing", "nobody"). Standalone morphemes most often appear uncompounded, but most can also act as component morphemes, whether this is through compounding with roots and other standalone morpheme, or adding a word ending. An example of standalone compounding is the word "dudekjara" ("twenty-year"), which contains the standalone morphemes "du" ("two") and "dek" ("ten"), the root "jar" ("year"), and the word ending "a" (adjective). The word "adiaŭi" ("to say goodbye") is formed using the standalone morpheme "adiaŭ" ("goodbye") and the word ending "i" (infinitive verb).

Forming compound words is a relatively permissive process. Fundamento de Esperanto, the official guide to Esperanto grammar, specifies only the basic mechanism for compound word formation (Zamenhof, 1905). Compound words are always formed by morpheme juxtaposition, and the principle morpheme occurs at the end of a word. For example, "ŝipvaporo" means "steam from a ship", while "vaporŝipo" means "steamship" (both words contain the morphemes "vapor" ("steam") and "ŝip" ("ship")). Roots can either be directly juxtaposed or separated by a word ending ("vaporŝipo" and "vaporoŝipo" are equivalent in meaning). The most common word ending to occur in the middle of words is "o", but the uninflected word endings "a", "i", "e" also often occur, as well as the accusative adverb ending "en". A word must always end with a word ending or a standalone morpheme, never with a root.

3. Previous Work

3.1. Other Agglutinative Languages

Koskenniemi (1984) proposed an influential morphological analysis model, the so-called "two-level morphology", which is applicable to languages with various morphologies, including agglutinative. The model consists of two components: a lexicon and a set of rules. The lexicon is a predefined list of tagged morphemes, and the rules are a set of finite state transducers, which directly transform an input word into a list of tagged component morphemes.

The ideas used by Koskenniemi (using a set of categorized morphemes and representing morphological rules as a finite state model) have proved to be a useful starting point for many subsequent studies. Alegria et al. (1996) developed a morphological analysis pipeline for Basque, directly incorporating Koskenniemi's model. Other studies have incorporated statistical finite-state models, such as Markov models or conditional random fields, for disambiguation. Rios and Mamani (2014) implemented a morphological analysis system for the Quechua language, using finite state transducers to recognize possible morphological analyses, and conditional random fields to perform disambiguation. Hakkani-Tür et al. (2002) performed Turkish morphological disambiguation using hidden Markov models.

Depending on language-specific considerations, it is potentially useful to incorporate rule-based analysis steps that do not necessarily fit a finite-state model. Ezeiza et al. (1998) used a combination of constraint grammar rules and a hidden Markov model to disambiguate morpheme part-of-speech tags in Basque words. Nongmeikapam et al. (2012) performed morphological segmentation for Manipuri, incorporating Manipuri syllabification rules and an n-gram Markov model. Solak and Oflazer (1992) implemented a spelling checking system for Turkish using various phonological and morphological rules. The first segmentation found (via maximal morpheme matching) that follows these rules is accepted.

Like many of these previous approaches, I apply Koskenniemi's general approach to Esperanto. Morphemes are classified by part-of-speech and semantic properties, and an n-gram Markov model is used for disambiguation.

3.2. Esperanto

Some morphological analysis methods have been developed for Esperanto, but this is still a largely unexplored topic.

McBurnett (1985) wrote a morphological segmentation algorithm, which maximizes the lengths of morphemes as a word is scanned from left to right, incorporating a few rules to ensure a grammatically legal segmentation is found. For example, the accusative and plural markers must occur in a specific order after a word ending morpheme, and a word cannot end with a root or affix. Maximal morpheme matching has been incorporated into morphological analysis systems for other agglutinative languages, including German (for compound nouns only) (Lezius et al., 1998) and Turkish (Solak and Oflazer, 1992). Thus, it is valuable to directly compare McBurnett's approach to other approaches of Esperanto morphological segmentation.

Hana (1998) developed a two-level morphology system for Esperanto by descriptively analyzing word formation patterns. This system was able to recognize most Esperanto words in a corpus, and reported 13.6% morphological ambiguity.

Some Esperanto spell checkers use morphological considerations. Esperantilo is an application that contains a spell checker along with many other linguistic tools for Esperanto (Trzewik, 2006). The spell checker uses a list of base morphemes, each with a set of prefixes, suffixes, and word endings that are often used with the base morpheme. A word is evaluated using rule-based criteria, which ultimately limits the complexity of a word relative to known derivations. Blahuš (2009) used the Hunspell framework to write a spell checker for the open source word processor OpenOffice. This spell checker was implemented using pattern matching based on known, fixed-length morpheme combinations, where morphemes were categorized by semantic and part-of-speech properties. Although both of these systems work well for many words, neither fully encapsulates the agglutinative nature of Esperanto morphology.

This study attempts to construct an algorithm that can segment any Esperanto word, which requires the ability to process words with any number of morphemes. McBurnett's and Hana's approaches are directly applicable to this goal, though this study focuses on developing a statistical approach. I do experiment with adding a simple rule-based step, where some non-grammatical segmentations are discarded before disambiguation, though this is much less sophisticated than Hana's system.

4. Methods

This approach[1] focuses on using a modified n-gram Markov model for disambiguation, where states represent semantic and part-of-speech classes of morphemes. Various orders of n-gram Markov models were tried, as it is not immediately evident which value of n would be optimal.

In addition, I implemented a maximal morpheme matching algorithm, which uses a simple rule-based step that discards ungrammatical segmentations before disambiguation, similar to McBurnett's approach.

To evaluate the results of each disambiguation method, I compare the segmentation accuracy to the expected accuracy if a random valid segmentation is chosen.

For all outputs, only segmentation accuracy is reported, as opposed to tagging accuracy, as only a set of presegmented words was readily available. However, this is not a huge disadvantage, since most morphemes only belong to one class, as defined in this study.

Additionally, this method does not attempt to perform hierarchical disambiguation of morphological structure, e.g. determining whether to interpret "unlockable" as "[un+lock]able" ("able to unlock"), or as "un[lock+able]" ("not able to lock"). A hierarchical disambiguation step can be applied independently after segmentation, and for many applications, assuming a linear morphological structure may be sufficient.

[1] Source code available at `https://github.com/tguinard/EsperantoWordSegmenter`

General Category	Tags
Standalone	Adverb, Article, Conjunction, Correlative, Exclamation, Number, Preposition, Pronoun
Affix	AdjectiveSuffix, NounSuffix, NumberSuffix, PeopleAnimalSuffix, TenseSuffix, VerbSuffix, NounPrefix, PeopleAnimalPrefix, PrepositionPrefix, VerbPrefix
Root	Adjective, Adverb, Noun, PeopleAnimal, Verb
Mid-Word Endings	O, OtherMidWordEndings
Word Endings	AdjectiveEnding, AdverbEnding, NounEnding, PronounCorrelativeEnding, VerbEnding

Table 1. Morpheme Categorization

4.1. Datasets

4.1.1. Lexicon

All roots that occur in Esperantilo (Trzewik, 2006) were used, as well as all standalone and affix morphemes from Akademio de Esperanto (2008).

Akademio de Esperanto lists prefixes, suffixes, and standalone morphemes separately. I manually categorized standalone morphemes based on part of speech. Prefixes and suffixes were manually categorized by which kind of morphemes they often modify, barring two exceptions. Tense suffixes, used to create participles in compound tenses, were differentiated from verb suffixes. The preposition prefix class consists of morphemes that can act as either prepositions or prefixes.

Roots were categorized by part of speech, using the associated word endings provided by Esperantilo. I used one additional semantic class for roots: people and animals. I defined this class as any noun morpheme that can use the suffix “in” (which makes a word feminine). These morphemes were removed from the noun category.

Word endings were categorized manually by part of speech and whether the morpheme can be used in the middle of a word. Although the plural and accusative markers (“j” and “n”) are considered separate morphemes, all possible combinations of word endings, the plural marker, and the accusative marker were explicitly listed. For example, “o”, “oj”, “on”, and “ojn” were all listed as separate morphemes. However, the plural and accusative markers are also listed separately since they may modify pronouns and correlatives; the Markov model training set should only list the plural and accusative markers as separate morphemes in this case.

An overview of the tags used can be found in Table 1.

4.1.2. Training and Testing Sets: Presegmented Dictionary Words

The ESPSOF project lists over 50,000 Esperanto words segmented into component morphemes (Witkam, 2008). The word list was constructed from various Esperanto dictionaries, and the segmentations were manually adjusted by Witkam. Only a subset of this list is used as input for this study since not all of the roots used in ESPSOF are listed in Esperantilo.

The total size of this input set is 42,356 words, which were split into a training set and test set (respectively used to set and test the Markov model parameters). Three-quarters of the words were used in the training set, and one-quarter in the test set. This three-quarters split was held over words with a consistent number of morphemes (e.g. three-quarters of words with two morphemes are in the training set). For all experiments run in this study, the same test set and training set were used.

Setting the Markov model parameters requires these segmentations to be tagged. Most morphemes belong to only one class as defined in this study, but for those that belong to multiple classes, simple rules are applied to determine the correct tag. For example, roots and word endings should match in part of speech if possible. If there is still uncertainty in the correct tag to assign, all possible tags are used with equal weight, but the total influence of each word on the Markov model is equal.

4.2. Segmentation Algorithm with Markov Model

There are two steps to the segmentation algorithm: finding all possible segmentations using a trie lookup algorithm, then selecting the best segmentation using a Markov model.

4.2.1. Segmentation

The segmentation phase finds all morpheme sequences that form the input word when juxtaposed. During this step, a minimalistic set of rules may be optionally applied:

- A word cannot end with a root or affix.
- The accusative marker "n" and the plural marker "j" can only appear after pronouns or correlatives (or after some word endings, but this is built into the lexicon).
- The definite article "la" cannot be combined with other morphemes.

All morpheme sequences are found via trie lookup.

For the ESPSOF word list, when the rules are applied, a word has a mean of 2.15 segmentations, 53.5% of words have at least two possible segmentations, and the largest number of distinct segmentations is 112. Thus, disambiguation is necessary.

4.2.2. Disambiguation

Disambiguation is performed using a modified n-gram Markov model. Each state represents n morpheme classes.

For the unigram model, each traversal begins on a state called "Start", visits the states corresponding to each morpheme class, and finishes on a state called "End". For example, in the segmentation "kat'okul'o", the individual morphemes are Esperanto for "cat", "eye", and (noun ending). The sequence of states visited is:

$$Start \rightarrow PeopleAnimal \rightarrow Noun \rightarrow NounEnding \rightarrow End$$

The frequency of each transition in the training set is used to calculate probabilities used by the Markov model.

The probability that the current state is B, given that the previous state was A, or $P(B|A)$, is related to the frequency of transitions from A to B, or $|(A, B)|$, and the sum of the frequency of transitions from state A to any state, S, or $|(A, S)|$.

$$P(B|A) = \frac{|(A, B)|}{\sum_{S \in States} |(A, S)|}$$

The score of the traversal, T, is calculated as follows. $|new\_class(B)|$ is the number of morphemes represented the last morpheme class in state B's n-gram, and α is a positive real number. For each word, the segmentation with the highest score is accepted as the correct segmentation. Occasionally, more than one segmentation may share the highest score. If this is the case, the ambiguity is resolved via maximal morpheme matching.

$$score(T) = \prod_{(A,B) \in T} \frac{\alpha \cdot P(B|A)}{|new\_class(B)|}$$

If α is omitted, this forms a straightforward Markov model, adjusted for unequal morpheme class sizes. Including α changes how often longer morpheme sequences are preferred. An optimal value for α can be found empirically in the training set.

For the bigram Markov model, each state represents two consecutive tags, and for the trigram Markov model, each state represents three consecutive tags. The beginning state always represents n Start tags. For example, the transition sequence of "kat'okul'o" for the bigram model is:

$$(Start \cdot Start) \rightarrow (Start \cdot PeopleAnimal) \rightarrow (PeopleAnimal \cdot Noun) \rightarrow$$
$$(Noun \cdot NounEnding) \rightarrow (NounEnding \cdot End)$$

For all models, the score calculation is equivalent, including the value of α (the number of states is constant between models for a given segmentation).

4.3. Additional Tests

4.3.1. Maximal Morpheme Match

This algorithm uses the same segmentation phase as the Markov model approach, but then selects the segmentation where the initial morphemes are as long as possible. That is, the length of the first morpheme is maximized, and if there is still ambiguity, the length of the subsequent morpheme is maximized, and this is repeated until there is no ambiguity.

The performance of this algorithm was compared with the Markov models by running this algorithm on all words from the ESPSOF word list (i.e. both the training set and the test set).

4.3.2. Randomly Selecting a Segmentation

As a baseline for comparing accuracy, I calculated the expected accuracy of randomly selecting a segmentation after the initial segmentation phase. This was applied to all words from the ESPSOF word list.

5. Results

When evaluating segmentation accuracy, a segmentation is considered correct if it equivalent to the expected segmentation, with one exception: the output segmentation contains a morpheme that appears in Esperantilo but not in ESPSOF, and this morpheme can be constructed from multiple morphemes in the expected solution. By inspecting the output, this is caused by Esperantilo listing morphemes that could be considered the combination of several morphemes. As an example, ESPSOF segments "prezidanto" ("president") as "prezid'ant'o" ("someone who presides"), while Esperantilo lists "prezidant" as a separate morpheme, so the output segmentation is "prezidant'o".

5.1. Various n-gram Markov Models

The segmentation accuracies of the three Markov models, with no rule-based step, are shown in Table 2. Although accuracies of the Markov models are high overall, there is a definite decrease in accuracy as the number of morphemes per word increases. All three models perform very similarly, though the higher order n-gram models are slightly more accurate overall.

This approach implements maximal morpheme matching as a secondary line of disambiguation in the case that multiple segmentations share the same highest score (this happened about 0.2-0.3% of the time). Depending on the model and word set, this strategy correctly resolved between 61-76% of these ambiguities.

Number of Morphemes	1	2	3	4	5	6	7	Any
Percent of Input Words	0.378	30.1	47.3	19.3	2.81	0.168	0.0142	100
Unigram: Training Set	1.00	1.00	0.990	0.966	0.909	0.811	0.750	0.986
Unigram: Test Set	1.00	0.999	0.989	0.963	0.906	0.944	1.00	0.985
Bigram: Training Set	1.00	1.00	0.992	0.971	0.936	0.906	1.00	0.989
Bigram: Test Set	1.00	1.00	0.991	0.969	0.923	0.833	0.500	0.989
Trigram: Training Set	1.00	1.00	0.992	0.971	0.933	0.962	1.00	0.989
Trigram: Test Set	1.00	1.00	0.991	0.973	0.916	0.833	0.500	0.987

Table 2. Markov model segmentation accuracies (no rules applied)

In terms of the errors that did occur, I observed that some were due to the inconsistent segmentation technique present in the ESPSOF word list. For example, ESPSOF segments the correlative “nenio” (“nothing”) as a root and word ending in “neni’o’far’ul’o” (“person who does nothing”), but other correlatives are treated as standalone morphemes, such as “nenies” (“nobody’s”) in “nenies’land’o” (“no man’s land”). Additionally, ESPSOF segments “esperanto” as “esper’ant’o” (“one who hopes”), which is the original etymology of the language’s name, but “esperant” is used as a single morpheme elsewhere in the list. These inconsistencies seem to account for approximately 10% of the total errors for each model.

In the test sets of each model, the erroneous segmentations had the same number of morphemes as the ESPSOF segmentation 57-61% of the time. The erroneous segmentations had too few morphemes 35-41% of the time and too many morphemes 2-4% of the time.

For the segmentations that had too few morphemes, most of the errors were common between all three models in the test set. 49 of these errors were common between all three models, while the unigram model had the most such errors (57). For all models, 72-76% of these erroneous segmentations combined two morphemes from ESPSOF’s solution to form a single morpheme. For example, “hufofero” should be segmented as “huf’o’fer’o” (“horseshoe”, literally “hoof iron”), but each model produced “huf’ofer’o”, (“hoof offer”). This type of error seems tricky to overcome, especially when the merged morpheme has a similar semantic class to the two separated morphemes.

There were very few instances where a segmentation with too may morphemes was produced, but this occurred most often in the unigram model (6 errors, vs. 3 each for the bigram and trigram models). The extra errors for the unigram model were due to overfavoring the accusative "n" morpheme. For example, "vinmiksaĵo" should be segmented as "vin'miks'aĵ'o" ("wine mixture"), but the unigram model produced "vi'n'miks'aĵ'o" (nonsense, literally "you (accusative) mixture").

The majority of the variation between the three models came from instances where the segmentation produced had the same number of morphemes as expected. There were 100 such errors for the unigram model, 82 for the bigram, and 76 for the trigram. 50 of these errors were common between all three models. These errors most directly show where Esperanto morphology does not follow a specific n-gram model, as the α factor does not influence these errors. For example, the unigram model erroneously uses mid-word endings more often than the bigram and trigram models, e.g. "help'a'gad'o" ("helpful cod") instead of "help'ag'ad'o" ("acting helpful").

Some of the errors that were not caused by inconsistencies in ESPSOF's segmentation may be resolved by improving the tag set. The presented morpheme categorization was effective, but optimal categorization is still an open issue.

5.2. Comparison with Maximal Matching and Random Selection

Table 3 compares the unigram Markov model with the maximal morpheme matching algorithm and the random selection strategy.

In terms of overall accuracy, the Markov model is significantly more accurate than maximal matching, though both developed algorithms are significantly more accurate than randomly choosing a segmentation.

The accuracy of the random selection method notably decreases as the number of morphemes increases, so it is natural for any segmentation algorithm to perform worse as the number of morphemes per word increases.

The maximal matching's performance is much more sensitive to the number of morphemes per word than the Markov model is. For words with only two morphemes, maximal matching performs comparably to the Markov model, but the accuracy quickly drops as the number of morphemes increases, approaching the accuracy of the random selection method.

When adding the rule-based step to the Markov models, the performance only changed for the test set of the unigram and trigram models, which correctly segmented one and two additional words respectively. However, adding rules significantly improves the accuracy of the maximal matching and random selection methods, as seen in Table 3.

Number of Morphemes	1	2	3	4	5	6	7	Any
Percent of Input Words	0.378	30.1	47.3	19.3	2.81	0.168	0.0142	100
Unigram: Training Set	1.00	1.00	0.990	0.966	0.909	0.811	0.750	0.986
Unigram: Test Set	1.00	0.999	0.989	0.963	0.906	0.944	1.00	0.985
Maximal Matching	1.00	1.00	0.970	0.833	0.676	0.577	0.333	0.944
Maximal Matching: No Rules	1.00	0.995	0.948	0.801	0.638	0.535	0.333	0.925
Random Selection	0.902	0.709	0.685	0.623	0.538	0.428	0.412	0.676
Random Selection: No Rules	0.750	0.630	0.541	0.437	0.330	0.202	0.208	0.542

Table 3. Comparison of Markov model, maximal matching, and random selection segmentation accuracies

6. Conclusion

This study investigated an n-gram Markov model approach to Esperanto morphological segmentation, as well as a maximal matching approach for comparison. An extra factor was added to the Markov model to adjust how often longer sequences of morphemes are accepted. Morphemes were categorized by part of speech, with a few extra subclasses, which was sufficient for producing a high segmentation accuracy.

There was not much difference between the performances of the various n-gram orders, although the bigram and trigram models were slightly more accurate for both the training and test sets. Both the Markov model and maximal matching approaches performed significantly better than randomly selecting a valid dissection, but the Markov model is more scalable to words with more morphemes. The rule-based step used in this study was useful for improving the accuracy of the maximal matching algorithm, but had no significant impact on the Markov model performances.

Bibliography

Agirre, Eneko, Inaki Alegria, Xabier Arregi, Xabier Artola, A Díaz de Ilarraza, Montse Maritxalar, Kepa Sarasola, and Miriam Urkia. XUXEN: A spelling checker/corrector for Basque based on Two-Level morphology. In *Proceedings of the third conference on Applied natural language processing*, pages 119–125. Association for Computational Linguistics, 1992.

Akademio de Esperanto. Akademia Vortaro, 2008. URL `http://akademio-de-esperanto.org/akademia_vortaro/`.

Alegria, Iñaki, Xabier Artola, Kepa Sarasola, and Miriam Urkia. Automatic morphological analysis of Basque. *Literary & Linguistic Computing*, 11(4):193–203, 1996.

Blahuš, Marek. Morphology-Aware Spell-Checking Dictionary for Esperanto. *RASLAN 2009 Recent Advances in Slavonic Natural Language Processing*, page 3, 2009.

Ezeiza, Nerea, Iñaki Alegria, José María Arriola, Rubén Urizar, and Itziar Aduriz. Combining stochastic and rule-based methods for disambiguation in agglutinative languages. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics-Volume 1*, pages 380–384. Association for Computational Linguistics, 1998.

Goldwater, Sharon and David McClosky. Improving statistical MT through morphological analysis. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 676–683. Association for Computational Linguistics, 2005.

Hakkani-Tür, Dilek Z, Kemal Oflazer, and Gökhan Tür. Statistical morphological disambiguation for agglutinative languages. *Computers and the Humanities*, 36(4):381–410, 2002.

Hana, Jiří. Two-level morphology of Esperanto. Master's thesis, Charles University Prague, Faculty of Mathematics and Physics, 1998. URL `http://www.ling.ohio-state.edu/~hana/esr/thesis.pdf`.

Kalocsay, Kálmán and Gaston Waringhien. *Plena Analiza Gramatiko de Esperanto*, volume 2. Universala Esperanto-Asocio, 1985.

Koskenniemi, Kimmo. A general computational model for word-form recognition and production. In *Proceedings of the 10th international conference on Computational Linguistics*, pages 178–181. Association for Computational Linguistics, 1984.

Lee, Young-Suk. Morphological analysis for statistical machine translation. In *Proceedings of HLT-NAACL 2004: Short Papers*, pages 57–60. Association for Computational Linguistics, 2004.

Lezius, Wolfgang, Reinhard Rapp, and Manfred Wettler. A freely available morphological analyzer, disambiguator and context sensitive lemmatizer for German. In *Proceedings of the 17th international conference on Computational linguistics-Volume 2*, pages 743–748. Association for Computational Linguistics, 1998.

McBurnett, Neal. Ilaro por Esperantaj Redaktoroj (Ilaro - a Toolkit for Esperanto Editors), 1985. URL `http://bcn.boulder.co.us/~neal/`.

Nongmeikapam, Kishorjit, Vidya Raj, Rk Yumnam, and Nirmal Sivaji Bandyopadhyay. Manipuri Morpheme Identification. *24th International Conference on Computational Linguistics*, pages 95–107, 2012.

Rios, Annette and Richard Castro Mamani. Morphological Disambiguation and Text Normalization for Southern Quechua Varieties. *COLING 2014*, page 39, 2014.

Solak, Aysin and Kemal Oflazer. Parsing agglutinative word structures and its application to spelling checking for Turkish. In *Proceedings of the 14th conference on Computational linguistics-Volume 1*, pages 39–45. Association for Computational Linguistics, 1992.

Trzewik, Artur. Esperantilo - text editor with particular Esperanto functions, spell and grammar checking and machine translation, 2006. URL `http://www.xdobry.de/esperantoedit/index_en.html`.

Witkam, Toon. ESPSOF (Esperanto-Softvaro), 2008. URL `http://www.espsof.com/`.

Zamenhof, L. L. Fundamento de Esperanto, 1905. URL `http://akademio-de-esperanto.org/fundamento/gramatiko_angla.html`.

Address for correspondence:
Theresa Guinard
`tguinard@gmail.com`
16801 NE 39th Ct #F2020
Redmond, WA, USA 98052

The Prague Bulletin of Mathematical Linguistics
NUMBER 105 APRIL 2016 77–99

A Comparison of Four Character-Level String-to-String Translation Models for (OCR) Spelling Error Correction

Steffen Eger[a], Tim vor der Brück[b], Alexander Mehler[c]

[a] Ubiquitous Knowledge Processing Lab, Technische Universität Darmstadt
[b] CC Distributed Secure Software Systems, Lucerne University of Applied Sciences and Arts
[c] Text Technology Lab, Goethe University Frankfurt am Main

Abstract

We consider the isolated spelling error correction problem as a specific subproblem of the more general string-to-string translation problem. In this context, we investigate four general string-to-string transformation models that have been suggested in recent years and apply them within the spelling error correction paradigm. In particular, we investigate how a simple 'k-best decoding plus dictionary lookup' strategy performs in this context and find that such an approach can significantly outdo baselines such as edit distance, weighted edit distance, and the noisy channel Brill and Moore model to spelling error correction. We also consider elementary combination techniques for our models such as language model weighted majority voting and center string combination. Finally, we consider real-world OCR post-correction for a dataset sampled from medieval Latin texts.

1. Introduction

Spelling error correction is a classical and important natural language processing (NLP) task, which, due to the large amount of unedited text available online, such as in tweets, blogs, and emails, has become even more relevant in recent times. Moreover, spelling error correction, in a broader meaning of the term, has also been of interest in the digital humanities where, for instance, large amounts of OCR (Optical character recognition) scanned text of historical or contemporary documents must be post-processed, or, even more generally, normalized (Mitankin et al., 2014; Springmann et al., 2014). In the same digital humanities context, spelling error correction

 Corresponding author: eger@ukp.informatik.tu-darmstadt.de
Cite as: Steffen Eger, Tim vor der Brück, Alexander Mehler. A Comparison of Four Character-Level String-to-String Translation Models for (OCR) Spelling Error Correction. The Prague Bulletin of Mathematical Linguistics No. 105, 2016, pp. 77–99. doi: 10.1515/pralin-2016-0004.

may be important in correcting errors committed by scribes in reproducing historical documents (Reynolds and Wilson, 1991). Beyond error correction, one faces wide ranges of co-existing spelling variants especially in documents of historical languages (e.g., medieval Latin) that must be normalized/standardized in order to be finally mapped to their corresponding lemmas.

Approaches to spelling correction (or standardization) are typically distinguished as to whether they target *isolated word-error correction* or *context-sensitive spelling correction* — sometimes also called *real-world spelling error correction* — in which errors may be corrected based on surrounding contextual word information. Many spelling error correction models have been suggested in the literature, among them, and most famously, the (generative) noisy channel model (Brill and Moore, 2000), discriminative models (Okazaki et al., 2008), finite-state techniques, as well as a plethora of local improvements and refinements for each class of models. In this work, rather than primarily suggesting new models for spelling error correction, we compare *general string-to-string translation* models developed in NLP contexts – typically, however, not within the area of spelling error correction — and survey methods for combining the outputs of the systems. The models we investigate have the following characteristics:

- They are *character-level*, that is, corrections are learned and implemented at the character-level, ignoring contextual words. Accordingly, in this work, our main focus is on isolated word-error correction, which may be considered harder than the context-sensitive spelling correction problem since surrounding contextual word cues are not available.[1] However, our experiments also include a real-world error correction setting.
- The models we survey are *general* in that they are not restricted to the spelling error correction task but can also be applied to many problems which require string-to-string translations, such as grapheme-to-phoneme conversion, transliteration, lemmatization, and others.[2] We think that generality and transferability of a model (in conjunction with accuracy) are central criteria of its quality.
- The models are *learned from data*, and in particular, are trained on pairs of strings of the form (x, y) where x is a misspelled word and y a desired correction.

The four approaches we survey are the Sequitur string-to-string translation model (Bisani and Ney, 2008), DirecTL+ (Jiampojamarn et al., 2010a), the contextual edit distance model suggested in Cotterell et al. (2014), and a model adaption of Eger (2012) which we call AliSeTra (Align-Segment-Translate). Although the first two models

[1] Also, one solution for real-world spelling error correction is to generate several candidates from an isolated spelling error correction model and then select the most likely candidate based on a word-level language model. In this sense, targeting isolated spelling error correction may be the first, and crucial, step in real-world spelling error correction.

[2] The only type of restrictions that our models make are *monotonicity* between input and output string characters, but otherwise allow, for instance, for many-to-many character relationships between input and output strings. This scenario is sometimes also referred to as *substring-to-substring* translation.

(and also the last) have been developed within the field of grapheme-to-phoneme (G2P) conversion and have been applied to related fields such as transliteration, too, their potential for the field of spelling error correction has apparently not yet been examined.[3]

We examine the suitability of the selected string-to-string translation models regarding the task of spelling error correction. To this end, we review the performance of these models and study the impact of additional resources (such as dictionaries and language models) on their effectiveness. Further, we investigate how to combine the output of the systems in order to get a system that performs as least as good as each of its component models. Note that combining string-valued variables is not a trivial problem since, for instance, the lengths of the strings predicted by the different systems may differ.

We show that by using k-best decoding in conjunction with a lexicon (dictionary), the string-to-string translation models considered here achieve much better results on the spelling correction task than three baselines, namely, edit distance, weighted edit distance and the Brill and Moore model. On two different data sets, three of the four models achieve word accuracy rates which are 5% resp. 25% better than the Brill and Moore baseline, which itself improves considerably upon edit distance and weighted edit distance. We also show that combining the models via language model weighted majority voting leads to yet another significant performance boost.

The article is structured as follows. In Section 2, we survey related work. In Section 3, we discuss the four string-to-string translation models and explain our techniques of combining them. Section 4 outlines the datasets used for evaluating these systems, viz., a Latin OCR dataset and a dataset of spelling errors in English tweets. Section 5, addresses three questions: (1) what are the accuracies of the four models on two different spelling correction data sets; (2) how can we improve the systems' performances by means of k-best output lists, language models and dictionaries; and (3) how well does the ensemble perform for different combination techniques — we consider weighted majority voting as well as center string ensembles. In Section 6, we touch upon the real-world spelling correction task, making use of our results in Section 5. In Section 7, we conclude.

2. Related Work

Brill and Moore (2000) suggest to solve the spelling error correction problem in the framework of the noisy channel model via maximizing the product of source model (language model) and the channel model for correcting a false input. Toutanova and Moore (2002) refine this model by integration of phonetic information. Cucerzan and Brill (2004) apply the noisy channel approach repeatedly, with the intent to cor-

[3]Similar investigations of G2P-inspired models for other tasks have been conducted, e.g., for lemmatization (Nicolai et al., 2015; Eger, 2015a).

rect more complex errors. More recent approaches to the spelling error correction problem include Okazaki et al. (2008), who suggest a discriminative model for candidate generation in spelling correction and, more generally, string transformation, and Wang et al. (2014), who propose an efficient log-linear model for correcting spelling errors, which, similar to the Brill and Moore (2000) model, is based on complex substring-to-substring substitutions. Farra et al. (2014) suggest a context-sensitive character-level spelling error correction model. Gubanov et al. (2014) improve the Cucerzan and Brill (2004) model by iterating the application of the basic noisy channel model for spelling correction in a stochastic manner.

Recently, there has been a surge of interest in solving the spelling error correction problem via the web (e.g., Whitelaw et al., 2009; Sun et al., 2010) and to correct query strings for search engines (e.g., Duan and Hsu, 2011, and many others). Further approaches to spelling correction include finite state techniques (e.g., Pirinen and Lindén, 2014) and deep graphical models (e.g., Raaijmakers, 2013). Kukich (1992) summarizes many of the earlier approaches to spell checking such as based on trie-based edit distances.

As mentioned, the models for spelling correction surveyed here are closely related to research on more general string-to-string transformation (translation) problems. This includes a variety of different models such as Cortes et al. (2005); Dreyer et al. (2008); Jiampojamarn et al. (2008); Bisani and Ney (2008); Cotterell et al. (2014); Wang et al. (2014); Sutskever et al. (2014); Novak et al. (2015).

3. Models

3.1. Alignment modeling

Two of the string-to-string translation systems evaluated below, DirecTL+ and AliSeTra, rely on *alignments* between input and output sequences (x, y). Since relationships between characters in spelling correction are typically of a complex nature as exemplified in Table 2, we assume that a *(monotone) many-to-many alignment* paradigm is the most suitable approach to modeling alignments in this scenario. We employ the monotone many-to-many aligner described in Jiampojamarn et al. (2007).[4] An implementation is available online at `https://code.google.com/p/m2m-aligner/`.

3.2. DirecTL+

DirecTL (Jiampojamarn et al., 2008, 2009) views string-to-string translation as a source sequence segmentation and subsequent sequence labeling task. The model extends its predecessor (Jiampojamarn et al., 2007) by folding the segmentation and

[4]This is an unsupervised many-to-many aligner. While supervised aligners are potentially more accurate (Eger, 2015b), the benefit of improved alignments for subsequent string transduction tasks is often marginal, particularly when training data is abundant.

tagging methods into a joint module. DirecTL+ (Jiampojamarn et al., 2010a) is a discriminative model for string-to-string translation that integrates joint n-gram features into DirecTL. The model has been applied in the context of grapheme-to-phoneme conversion (Jiampojamarn et al., 2010a) and in related domains such as transliteration (Jiampojamarn et al., 2010b). An online implementation is available at `https://code.google.com/p/directl-p/`.

3.3. Sequitur

Sequitur (Bisani and Ney, 2008) implements a joint n-gram model for string-to-string translation that, in the translation process from x to y, uses n-gram probabilities over pairs of substrings of the input and output sequence ('joint multigrams'). Duan and Hsu (2011) use a joint-multigram modeling, very much in the spirit of Sequitur, for query-string correction for search engines. A downloadable version of Sequitur is available at `http://www-i6.informatik.rwth-aachen.de/web/Software/g2p.html`.

3.4. AliSeTra

We develop our own model for string-to-string translation that, similarly to DirecTL+, treats string transduction as a sequence segmentation and subsequent sequence labeling task. In this approach, at *training time*, a sequence labeling model (in our case a discriminative conditional random field) is trained on many-to-many aligned data. Simultaneously, a sequence labeling module is trained for segmenting input sequences by ignoring the segmented y sequences in the aligned data, simply considering the segmented x sequences. We use a binary encoding scheme similarly as in Bartlett et al. (2008) and Eger (2013) for learning sequence segmentation. At *test time*, an input string x is segmented via the segmentation module and then the sequence labeling model is applied to obtain the output sequence. In contrast to DirecTL+, this approach ignores joint n-gram features and resorts to the pipeline approach to string-to-string translation. Its benefit is that it may be used in conjunction with any state-of-the-art sequence labeling system, so it may directly profit from improvements in tagging technology. We use CRF++ as a sequence labeler.[5] We call this model AliSeTra (Align-Segment-Translate). In Table 1, we illustrate its decoding phase and show sample aligned training data on which the sequence labeling models in AliSeTra are trained.

3.5. Contextual Edit Distance

Cotterell et al. (2014) design a discriminative string-to-string translation model where $p(y|x)$ is modeled via a probabilistic finite state transducer that encodes weighted edit operations transforming an input string x into an output string y (weighted

[5] Downloadable from `https://code.google.com/p/crfpp/`.

li-a-b-i-t-o	h-a-b-i-t-o	adliuc	$\rightsquigarrow$	a-d-li-u-c
a-d-j-u-t-o-r-i-u-ni	a-d-j-u-t-o-r-i-u-m			$\downarrow\downarrow\downarrow\downarrow\downarrow$
p-c-r-c-e-p-i-t	p-e-r-c-e-p-i-t			a-d-h-u-c

Table 1. Latin OCR spelling errors and their corrections. Left: Sample monotone many-to-many aligned training data, as obtained from the alignment procedure discussed in text. Alignment of characters indicated by dashes ('-')), one alignment per line. Right: AliSeTra *at test time. A new input string,* adliuc, *is first segmented into* a-d-li-u-c, *via a segmentation module trained on the segmented x strings in the training data. Then a tagging model, trained on the monotone many-to-many aligned pairs of (x, y) strings, assigns each (multi-)character in the segmentation its label, which can be a character or a multicharacter. This yields the predicted correction* adhuc *('hitherto').*

edit distance). Moreover, in their design, edit operations may be conditioned upon input and output context,[6] thus leading to a *stochastic contextual edit distance* model. An implementation is available from `http://hubal.cs.jhu.edu/personal/`.[7]

3.6. Baseline methods

As baseline methods for comparison, we use

- *edit distance* with the operations of insertion, deletion, and substitution as well as swapping of adjacent characters. That is, for a falsely spelled input x, this measure determines the string y in a dictionary whose edit distance to x is lowest;
- *weighted edit distance*, in which the weight of edit operations is learned from data (we use the above named many-to-many aligner with edit operations restricted appropriately to induce training sets) rather than set exogenously;[8]
- and the Brill and Moore model (Brill and Moore, 2000), which embeds a substring-to-substring translation model into a generative noisy channel framework. In this, the channel probability $p(x|y)$ is determined via (maximizing over) unigram models on substring segmentations of the form $\prod_i p(x_i|y_i)$, whereby

[6]The context is the preceding and subsequent characters in a string, not, e.g., the preceding words.

[7]The contextual edit distance model as designed in (Cotterell et al., 2014) is a locally normalized model suffering from the "label bias" problem and thus, potentially inadequate for our task. Although it has been primarily designed for incorporation in a Bayesian network over string-valued variables (Cotterell et al., 2015), we nonetheless include it here for comparison.

[8]In addition, we weight suggestions $\hat{y}$, for an input x, by a unigram word-level language model, which improves performance, as we found. The language model is trained on the same data sets as the language model for the Brill and Moore (2000) model; see below.

$x_1 \cdots x_r$ and $y_1 \cdots y_r$ are joint segmentations (i.e., an alignment) of x and y.[9] For the Brill and Moore (2000) model, we employ unigram word-level language models as source models.[10]

All these baselines are *dictionary-based*, that is, they retrieve corrections y given in a predefined dictionary D, which is typically advantageous (see our discussion below), but may lead to errors in case of, e.g., low quality of D. For efficient decoding, we employ a *trie*-based search strategy for finding corrections y in all three baseline methods presented. For edit distance, in case of ties between corrections — distinct forms y with same edit distance to x — we choose the lexicographically smallest form as the suggested correction.

For the English spelling error data (see below), we use the freely available (rule-based) spell checker Hunspell[11] as a reference.

3.7. System combination

Since we investigate multiple systems for spelling correction, a natural question to ask is how the outputs of the different systems can be combined. Clearly, this is a challenging task, and different approaches, with different levels of sophistication, have been suggested, both within the domain of machine translation (Rosti et al., 2007) and the field of string transductions (see, e.g., Cortes et al. (2014) for a survey). In this work, where the main goal is the comparison of existing approaches, we resort to *simple* combination techniques illustrated below. For an input string x — a wrongly spelled or a wrongly OCR recognized word form — let $y_1, \dots, y_M$ denote the M predictions suggested by M different spelling correction systems. Then, we consider the following combination techniques:

- **Majority voting** chooses the most frequently suggested correction y among $y_1, \dots, y_M$.
- **Weighted majority voting**: here, each suggested correction y_ℓ receives a weight $w_\ell \in \mathbb{R}$, and the correction y among $y_1, \dots, y_M$ which maximizes $\sum_{\ell=1}^{M} w_\ell \mathbb{1}_{y_\ell = y}$ is chosen, where $\mathbb{1}_{a=b} = 1$ if $a = b$ and $\mathbb{1}_{a=b} = 0$ otherwise. We consider two weighting schemes:
  - *Accuracy weighted majority voting*: In this scheme, string y_ℓ receives weight w_ℓ proportional to the accuracy of system ℓ (e.g., as measured on a development set).

[9] In contrast, in Sequitur, for example, general n-*gram* models — rather than unigram models — over (x_i, y_i) pairs are used for modeling (joint) probabilities $p(x, y)$, indicating why Sequitur should typically outperform the Brill and Moore (2000) approach.

[10] For the Latin OCR data, as explicated below, these are trained on the Patrologia Latina (Migne, 1844–1855), and for the English Twitter data, the language model is based on a Wikipedia dump from 2013-09-04.

[11] `http://hunspell.sf.net`.

  - *Language model weighted majority voting*: In this scheme, suggestion y_ℓ receives weight w_ℓ proportional to the language model likelihood of string y_ℓ.
- **Center string decoding**: We define the center string among $y_1, \ldots, y_M$, as the string $y \in Y = \{y_1, \ldots, y_M\}$ whose average edit distance to all other strings in Y is minimized (Gusfield, 1997). A center string can be seen as an (efficient) approximation to the concept of a *consensus string* (Gusfield, 1997), which does not need to be in Y.

Clearly, a drawback of all our suggested combination techniques is that they can only select strings y that belong to $\{y_1, \ldots, y_M\}$. Hence, if none of the strings $y_1, \ldots, y_M$ is the true correction of the wrongly spelled form x, then the system combination prediction will also be wrong. A strength of our combination techniques is that they are easily and efficiently implementable and interpretable.

4. Data

We conduct experiments on two data sets. The first is a Latin OCR spelling correction data set, which we obtained by comparison of an OCR scan of a subpart of the Patrologia Latina (Migne, 1844–1855) with the original in electronic form. The second is a data set of spelling errors in tweets,[12] which we refer to as Twitter data set. For the Latin data, we automatically extracted pairs of strings (x, y), where x denotes a wrongly recognized/spelled OCR form and y its desired correction, via the Unix shell command `diff`, applied to the original text and its OCR scan. This yielded about 12,000 pairs of (x, y) strings. From this, we excluded all string pairs containing upper case or non-ASCII characters, as some of our systems could only deal with lower-case ASCII characters. This resulted in a much smaller (and cleaner) data set comprising $5,213$ string pairs. For the Twitter data, we took the first $5,000$ word pairs of the respective data set for testing and training. We removed two word pairs which contained underscores in the x strings, for the same reason as indicated above.

Table 2 illustrates some of the relationships between characters (or character subsequences) in Latin and English strings and their spelling corrections. As is well-known, in the field of classical spelling correction, as the Twitter dataset represents, errors are often driven by 'phonetic similarity' of characters representing sounds, such as *a/e*, *u/ou*, etc., or keyboard adjacency of the characters in question such as *n/m*, *c/v*, etc. In contrast, OCR spelling errors typically derive from the *visual* similarity of characters, such as *li/h*, *n/ra*, *t/l*, *i/j*, *in/m*, etc. As Table 2 also illustrates, more complex many-to-many relationships between characters of (x, y) pairs may not be uncommon; and they allow for a seemingly plausible interpretation of the processes underlying string transformations. For example, it seems plausible to assume that an OCR system mis-

[12] Available from `http://luululu.com/tweet/`.

takes *h* for *li*, rather than assuming that, for instance, it confuses *h* with *l* and subsequently inserts an *i*.

n $\rightarrow$ ra	p**n**eterea $\longrightarrow$ p**ra**eterea	mm $\rightarrow$ m	co**mm**ing $\longrightarrow$ co**m**ing
li $\rightarrow$ h	ad**li**uc $\longrightarrow$ ad**h**uc	n $\rightarrow$ m	victi**n** $\longrightarrow$ victi**m**
i $\rightarrow$ j	**i**uventam $\longrightarrow$ **j**uventam	t $\rightarrow$ th	**t**ink $\longrightarrow$ **th**ink
t $\rightarrow$ l	il**t**ustri $\longrightarrow$ il**l**ustri	u $\rightarrow$ ou	w**u**ld $\longrightarrow$ w**ou**ld
in $\rightarrow$ m	**in**isero $\longrightarrow$ **m**isero	a $\rightarrow$ e	emerg**a**ncy $\longrightarrow$ emerg**e**ncy
c $\rightarrow$ e	quoqu**c** $\longrightarrow$ quoqu**e**	c $\rightarrow$ v	ha**c**e $\longrightarrow$ ha**v**e

Table 2. Sample substring substitution patterns in Latin OCR data (left) and English Twitter data (right), indicated and in bold. The patterns were found via automatic alignment of string pairs.

5. Isolated error correction

System parametrizations

We run the four systems of Section 3 using the following parametrizations. For Sequitur, we train 7 successive models, where parameters are, in each case, optimized on a heldout 5% development set. For AliSeTra, we set the C constant in the CRF++ implementation, which determines over-/underfitting of the model, to the default value of 1. For k-best decoding, we employ a '$k_1 \times k_2$ strategy' for AliSeTra:[13] at test time, each string is segmented into the k_1 most likely segmentations, and then the sequence labeling model — we take as features all sequence m-grams that fit inside a window of size 5 centered around the current position in the segmented string — transduces each of these into the k_2 most likely corrected strings. Thus, this yields $k_1 \times k_2$ output string suggestions; we multiply the segmentation probabilities with the transduction probabilities to obtain an overall probability of a corrected string. Then, we re-sort the obtained corrections and keep the k most likely. For the DirecTL+ model, we choose, as context features, all m-grams inside a window of size 5 around the current position, as in the AliSeTra setting; we train a linear chain of order 1, set the joint multigram switch to 3 and the joint forward multigram switch to 1 (increasing the last three parameters did not seem to lead to better results, but only to longer runtimes). For Contextual Edit Distance, we choose the best-performing $(1, 1, 1)$ topology from the Cotterell et al. (2014) paper, which considers as context the previous and next x string characters and the previous y string character (the value of the backoff parameter is 0.5). In terms of training times on a 2.54 GHz processor, train-

[13]In all experiments, we set $k_1 = 5$ and $k_2 = 50$.

ing the first three models ran in several hours, across all folds, while the CONTEXTUAL EDIT DISTANCE model took days to train.

Evaluation setup

For the evaluation of our results, we employ 10-fold repeated random subsampling validation, in which, for each fold, we randomly split the data sets into training vs. test sets of size 90% vs. 10% of the whole data. Note that in random subsampling validation, training (as well as test) sets may overlap, across different folds.

Below, we indicate the performance of each of the four general string-to-string translation systems outlined in Section 3 in two different settings. In the *first setting*, we simply check whether the first-best string $\hat{y}$ predicted by a system S for an input string x matches y, the true correction for input string x. This is the typical evaluation scenario, e.g., in grapheme-to-phoneme conversion and related string-to-string translation fields such as transliteration. In an *alternative setting*, we let each system emit its k-best output predictions for an input string x, in decreasing order of (system-internal) probability, and then choose, as the system's prediction for x, the first-best string y^j, for $j = 1, \dots, k$, *that occurs in a predefined dictionary* D. If no string $y^1, \dots, y^k$ is in D, we choose y^1 as the system's prediction, as in the standard setting. Note that our first setting is a special case of the second setting in which $k = 1$.

Consulting a dictionary is done by most approaches to spelling correction. Combining a dictionary with k-best decoding in the manner described is apparently a plausible solution to integrating a dictionary in the setup of general string-to-string translation models. Note that our approach allows for predicting output strings that are not in the dictionary, which may be advantageous in case of low dictionary quality — but even if the quality of the dictionary is good, desired output strings may be missing (cf. Table 3).

For Latin, we choose a subset of ColLex.LA (Mehler et al., 2015) as our dictionary of choice and for English, we use ColLex.EN (vor der Brück et al., 2014).[14] Table 3 gives the number of entries in both lexicons as well as OOV numbers.

	Number of unique entries	OOV rate
Subset of ColLex.LA	4,269,104	$57/5213 = 1.09\%$
ColLex.EN	3,998,576	$189/4998 = 3.78\%$

Table 3. Dictionaries, their sizes, and OOV rates (number of corrections in each data set not in the dictionary).

[14] Both dictionaries are available from `http://collex.hucompute.org/`.

In both settings, we use **word accuracy** (WACC) as a performance measure, defined as the number of correctly translated strings over the total number of translated strings,

$$\text{WACC} = \frac{\sum_{i=1}^{n} \mathbb{1}_{\hat{y}_i = y_i \,|\, x_i}}{n},$$

where n is the size of the test set and $\mathbb{1}_{\hat{y}_i = y_i \,|\, x_i}$ is one or zero, depending on whether $\hat{y}_i = y_i$ or not (we use the | notation to indicate dependence of $y_i/\hat{y}_i$ on input x_i).[15]

5.1. Individual system results

Tables 4 and 5 list the results for the two data sets when using our above dictionary-based strategy with 1-best and 80-best decoding. Clearly, 80-best decoding yields much better results for all of the four methods, where word accuracy increases from about $16 - 70\%$ on the Latin OCR and $5 - 30\%$ on the Twitter data, relative to 1-best decoding, across all systems. This confirms that a dictionary may be very helpful in (OCR) spelling correction and that simple k-best decoding and first-best dictionary selection can be a good solution for integrating a dictionary into general string-to-string translation systems. In Figures 1 and 2, we plot each system's performance as a function of k in the k-best decoding strategy.

We also note that three of the four systems introduced in Section 3 — namely, AliSeTra, DirecTL+, Sequitur — have a very similar performance across the two data sets, whereas Contextual Edit Distance performs much worse, particularly in 1-best decoding. We attribute this to the fact that contextual edit distance considers much less context in our setup than do the other three systems.[16] Moreover, it operates on a single-character, rather than on a substring, or multi-character, level, which further reduces its contextual awareness.[17] However, we see that differences in system performances decrease as k increases. For example, for $k = 1$, the best system is approximately 60%/57% better than Contextual Edit Distance on the Latin OCR/Twitter data sets — while for $k = 80$, this reduces to 9%/28%. This indicates that Contextual Edit Distance may enumerate many of the relevant correct strings, for given input strings x, but has a higher chance of erring in correctly ranking them. We also note that the Twitter data set is apparently harder than the Latin OCR data set, as all systems exhibit worse performance on the former data set. This is, among other things,

[15] When an input x has multiple distinct translations in the test data set — e.g., *tis* ⟶ *this, is, its* — then, in the evaluation, we randomly choose one of these translations as the true translation. As discussed below, such cases happen relatively rarely. For example, in the Latin OCR data, 88.5% of all x forms have a unique correction associated with them, while in the Twitter data, this number is 61.5%.

[16] Increasing context size is critical, as the program's runtime is excessive. We did not experiment with larger context sizes for Contextual Edit Distance.

[17] Finally, contextual edit distance is locally normalized and thus suffers from the label bias problem as discussed earlier.

due to the fact that the Twitter data is generally more ambiguous than the Latin data in that an input string x is potentially related to more candidate alternatives.[18]

Model	1-best	80-best
AliSeTra	74.66 ± 1.26	87.33 ± 1.26
DirecTL+	$\mathbf{75.95 \pm 1.65}$	$\mathbf{88.35 \pm 1.54}$
Sequitur	73.67 ± 1.85	87.44 ± 1.90
Contextual Edit Distance	47.55 ± 1.77	81.12 ± 1.28
Edit distance		45.30 ± 2.04
Weighted edit distance		73.67 ± 1.21
Brill and Moore		84.20 ± 2.23

Table 4. Latin OCR data: Word accuracy in % for the k-best decoding strategy explicated in the text, and comparison with baseline methods; note, in particular, that we use a dictionary in conjunction with k-best decoding (1-best decoding is tantamount to ignoring the dictionary). The baseline methods are dictionary-based by their design, so the numbers simply indicate their word accuracy for their first-best prediction. In bold: Statistically indistinguishable best results (paired t-test, 5% level).

Model	1-best	80-best
AliSeTra	$\mathbf{68.38 \pm 1.52}$	$\mathbf{72.98 \pm 2.01}$
DirecTL+	$\mathbf{68.15 \pm 1.56}$	71.65 ± 2.12
Sequitur	63.01 ± 1.54	70.46 ± 1.60
Contextual Edit Distance	43.52 ± 2.28	56.78 ± 1.86
Edit distance		16.81 ± 1.78
Weighted edit distance		33.69 ± 2.11
Brill and Moore		58.08 ± 3.00
Hunspell		41.42 ± 1.96

Table 5. Twitter spelling correction data: Word accuracy in % for the k-best decoding strategy explicated in the text, and comparison with baseline methods.

[18] In the Latin OCR data, each x is on average associated with 1.0037 distinct y forms, while in the Twitter data, there are 1.1101 distinct y forms per x form. To illustrate, the possible corrections of *ot* in the Twitter data are *on, of, it, to, got, or, out*; similarly, *wat* may be corrected by *what, was, way, want, at*, etc. While in the evaluation, we remove this uncertainty by randomly assigning one of the strings as the correct output for a given input, at training time, this may lead to more inconsistency and ambiguity.

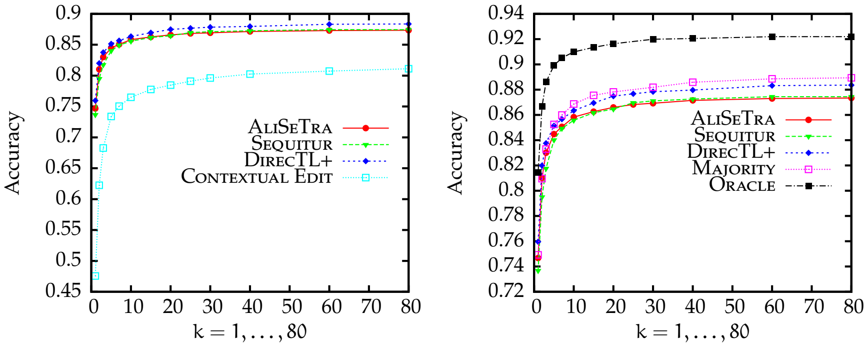


Figure 1. Latin OCR data, word accuracy as a function of k *in the* k*-best decoding strategy outlined in the text. Left: the four systems introduced in Section 3. Right: Three of the systems (excluding* Contextual Edit Distance, *for clarity) plus majority voting and oracle performance.*

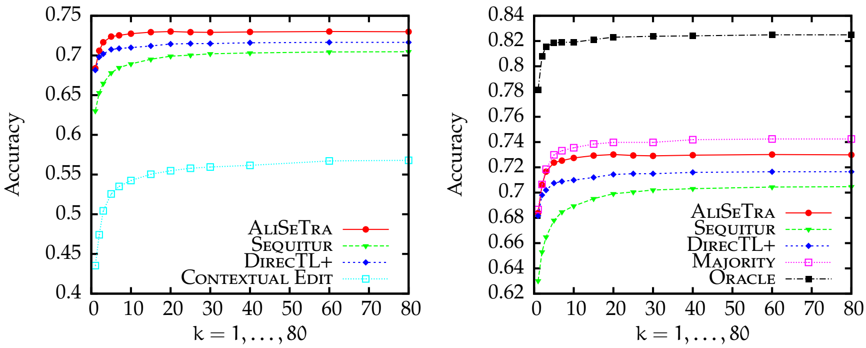


Figure 2. Twitter data, word accuracy as a function of k *in the* k*-best decoding strategy outlined in the text. Left: the four systems introduced in Section 3. Right: Three of the systems (excluding* Contextual Edit Distance, *for clarity) plus majority voting and oracle performance.*

Comparing the figures in the graphs and tables, we also see that three of the four general string-to-string translation systems surveyed perform much better than the baselines edit distance, weighted edit distance, and the Brill and Moore model. For instance, on the Latin OCR data, the best system is roughly 5% better than the performance of the Brill and Moore model, which itself is considerably better than edit distance or weighted edit distance, while on the Twitter data, this difference amounts to more than 25%. Oftentimes, the three of the four general string-to-string translation systems also perform on a level close to or above the level of the compared baselines, *even without using a dictionary*, as the 1-best results indicate.

In Figure 3, we provide another measure of system performance, *recall-at-*k. Under this measure, a system is correct for an input x if the true correction y is among the system's k-best predictions $y^1, \ldots, y^k$. Clearly, for fixed k, each system's performance under this measure must be at least as good as under the previous word accuracy measure for the k-best decoding strategy. Recall-at-k may be an important indicator for real-world spell checking, which often relies on a candidate generation module and a ranker for the candidates. Then, it may be sufficient for the candidate generation module to *generate* the true correction, as long as the ranker (often a word level n-gram model) can adequately discriminate between alternatives.

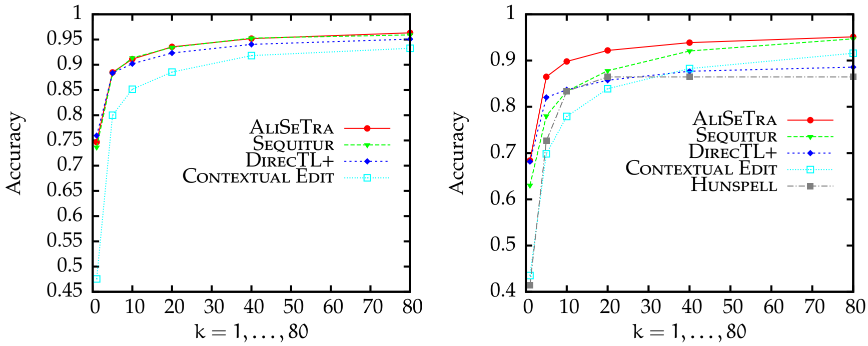


Figure 3. Recall-at-k *as described in the text. Left: Latin OCR data. Right: Twitter data.*

As seen in the figure, results are similar as in the previous setting — system performances increase significantly as k increases and system differences decrease in k. Interestingly, DirecTL+ appears to perform relatively worse under this measure than under the word accuracy measure, indicating that it seems to do a relatively better job in ranking alternatives, compared to the other systems. In contrast, Hunspell and Contextual Edit Distance, for example, which perform badly at predicting the exact true correction for an input, nonetheless appear relatively more capable of at least generating the true correction among their predictions. We also conclude that given that the recall-at-k of some of the systems is above 95% and 90% for the Latin OCR and Twitter data sets, respectively, while k-best decoding plus dictionary selection as outlined above yields word accuracy rates of (only) about 88% and 72%, respectively, our presented dictionary k-best decoding strategy could in principle be much improved upon.

5.2. System combination results

In Tables 6 and 7, we show results for the different system combination techniques outlined in Section 3. For the four systems surveyed in this work, we use the 80-best dictionary decoding strategy as outlined above as a basis for the system combination. We see that majority voting, center string combination, and weighted majority voting can increase performance significantly over the individual system accuracies. Majority voting gives slightly better results than center string combination. Even including weaker systems can be beneficial as the tables show. Typically, best results are obtained via integration of all systems except for (individually quite poor) standard edit distance. Compared to the individual systems, majority voting increases word accuracy by as much as 2% on the Latin OCR data set and as much as 5% on the Twitter data set; performance increases for center string combination are 1.1% and 3.5%, respectively.

Latin OCR				
Models	**Majority** (MV)	**Center String**	**Acc-MV**	**LM-MV**
A+D+S	88.52 ± 1.47	88.60 ± 1.50	$88.62^* \pm 1.40$	90.99 ± 1.32
+CED	88.93 ± 1.51	88.89 ± 1.48	$89.12^* \pm 1.45$	91.46 ± 1.16
+BM	89.82 ± 1.55	89.62 ± 1.39	$89.76^* \pm 1.22$	93.13 ± 0.92
+WE	$\mathbf{90.16 \pm 1.22}$	$\mathbf{89.93 \pm 1.36}$	$\mathbf{90.07^* \pm 1.33}$	$\mathbf{93.33 \pm 0.97}$
+ED	89.74 ± 1.45	89.33 ± 1.38	$89.80^* \pm 1.30$	93.27 ± 0.90

Table 6. Word accuracies for system combination techniques on Latin OCR data. Systems abbreviated by their first letters or initials (WE is weighted edit distance, ED is standard edit distance). In each column: statistically indistinguishable best results, paired t-test, 5% level. The results for accuracy-weighted majority voting are starred because we used the accuracies as obtained on the test data (usually, a development data set would need to be used for this), so that the results are 'upward' biased.

Accuracy-weighted majority voting does not typically result in large improvements over simple majority voting, if at all. Conversely, when we train a 10-gram character level language model (for Latin, on the original text from which the spelling correction (x, y) pairs were obtained; for Twitter, on the remaining roughly $35,000$ y strings that were not used in training/testing), and perform language model weighted majority voting, then this significantly increases results, by 3.5% on the Latin OCR data and 4.6% on the Twitter data, over standard majority voting combination.

English Twitter					
Models	**Maj.** (MV)	**Center Str.**	**Acc-MV**	**$\mathbf{LM_{Twitter}}$-MV**	**$\mathbf{LM_{Europarl}}$-MV**
A+D+S	74.56 ± 1.90	$\mathbf{75.08 \pm 2.05}$	$74.87^* \pm 2.05$	77.80 ± 1.59	$\mathbf{76.34 \pm 1.51}$
+CED	74.34 ± 2.24	$\mathbf{75.06 \pm 2.13}$	$75.03^* \pm 2.47$	78.28 ± 1.87	75.23 ± 1.62
+BM	76.09 ± 2.06	$\mathbf{75.23 \pm 2.31}$	$76.09^* \pm 2.38$	80.11 ± 1.93	74.20 ± 2.14
+WE	$\mathbf{76.69 \pm 1.79}$	$\mathbf{75.57 \pm 2.23}$	$\mathbf{76.69^* \pm 2.10}$	$\mathbf{80.58 \pm 1.94}$	72.49 ± 1.83
+ED	75.49 ± 1.89	74.31 ± 2.02	$\mathbf{76.69^* \pm 2.11}$	$\mathbf{80.69 \pm 1.98}$	72.56 ± 1.95

Table 7. Word accuracies for system combination techniques on English Twitter data.

Note that a language model may lead to deteriorations in results if being trained on data very dissimilar to the data on which it is to be applied and when weak systems are integrated into the majority voting process. For example, when we train a 10-gram character level language model on the English part of the Europarl corpus (Koehn, 2005), then language model weighted majority voting with 7 systems almost drops down to the word accuracy level of the single best system in the ensemble.

6. Real-world error correction

Finally, we consider the real-world spelling correction problem in our context, focusing on the Latin OCR data. To this end, we train two components: a spelling error correction model as outlined in the previous section and a language model (LM). We train the two most successful spelling correction systems from our previous setup — DirecTL+ and AliSeTra — on the previously described Latin OCR data,[19] this time not excluding word pairs containing upper-case or non-ASCII characters (so as to provide a 'real-world' situation). In addition, we train a 5-gram Kneser-Ney word-level LM via the SRILM toolkit[20] (Stolcke, 2002) on the union of the Patrologia Latina and the Latin Wikipedia data.[21] To combine the predictions of the LM and the discriminative string transducers, we opt for a power mean combination. In particular, for a potentially incorrect form x, we let the respective OCR post-corrector output its K-best (here, $K = 80$) suggestions $y_1, \ldots, y_K$. For the LM, we score each of these suggestions y by querying the LM on the sequence $x_{t-4} \cdots x_{t-1} y$, where x_{t-s} denotes the s-th word before word x at position t. Then, we choose the form $\hat{y}$ as the suggested correction which maximizes

$$\text{PM}\Big(\text{lm-score}(x_{t-4} \cdots x_{t-1} \hat{y}), \text{tm-score}(\hat{y}|x); w_{\text{LM}}, 1 - w_{\text{LM}}, p\Big)$$

[19] We keep 90% for training and 10% for testing.

[20] `http://www.speech.sri.com/projects/srilm/`

[21] Dump from 2015-10-10.

where lm-score denotes the LM score and tm-score denotes the score of the respective OCR transducer model. We normalize the scores such that they sum to 1 for all suggestions in the candidate list. Finally, $\mathrm{PM}(x, y; w_x, w_y, p)$ is the power mean $(w_x x^p + w_y y^p)^{1/p}$ where $w_x, w_y \geq 0$ with $w_x + w_y = 1$, and $p \in \mathbb{R}$. We consider here $p = 1$ (weighted arithmetic mean) and $p \to 0$ (weighted geometric mean); we refer to the latter case as $p = 0$, for convenience.

We consider two 'treatments', one in which we filter out suggestions $\hat{\mathbf{y}}$ not in the lexicon, and one in which no such filtering takes place. We consider a form $\mathbf{x}$ as potentially incorrect only if $\mathbf{x}$ is not in our Latin lexicon. When comparing the post-corrected text with the original, we face the problem that the two texts are not identical in that the original, e.g., contains additional text such as insertions introduced by the texts' editors ('[0026A]'). Thus, we find it easiest to measure the improvement between the scanned text version and our post-correction by applying the Unix `diff` command to the two files.[22]

Table 8 shows the results, for different values of w_{LM} and $p = 0, 1$. We note some general trends: using geometric averaging is always better than using arithmetic averaging, and using the DirecTL+ corrector is usually better than using AliSeTra, which is in accordance with the results highlighted in Table 4. Moreover, making the LM weight too large is typically detrimental; in these experiments, values $\leq 1/2$ were found to be best, indicating that the post-correctors typically perform better than the LM. Finally, using the lexicon as a filtering device has been beneficial in 8 out of 20 cases, but led to worse results in the remaining cases. A possible explanation is that, after filtering suggestions by whether they are contained in the lexicon, the candidates' LM and OCR corrector scores change since we renormalize them. Hence, if for example the LM has attributed a high score to an incorrect form this score may become even higher after filtering, thus leading to higher probability of a wrong selection. Finally, we note that the `diff` measure value between the original text and its scan is 1794, so our post-correction improves this value by roughly 28% (1294 and 1302 for AliSeTra and DirecTL+, respectively, in the best settings). While this seems to be a moderate improvement, we note that many wrongly scanned forms are in our lexicon; in particular, this concerned ligatures such as æ in the scan *memoriæ* of *memoriae*. Hence, these forms were not corrected at all since our correction addressed only forms not available in our lexicon.

7. Conclusion

We considered the isolated spelling error correction problem as a specific subproblem of the more general string-to-string translation problem. In this respect, we investigated four general string-to-string transformation models that have been suggested

[22] To be precise, our command for comparing the two versions is
`diff post-corrected.txt orig.txt -y |grep "|\|<\|>"|wc -l`.

	Lexicon	OCR corrector	0	1/4	1/2	3/4	1
$p = 1$	+	AliSeTra	1389	1376	**1366**	1439	1504
$p = 0$	+	AliSeTra	1389	1361	**1356**	1401	1504
$p = 1$	-	AliSeTra	1451	1390	**1339**	1407	1475
$p = 0$	-	AliSeTra	1451	1316	**1294**	1357	1475
$p = 1$	+	DirecTL+	1343	1336	**1330**	1406	1466
$p = 0$	+	DirecTL+	1343	**1325**	1330	1344	1466
$p = 1$	-	DirecTL+	1417	**1343**	1356	1412	1449
$p = 0$	-	DirecTL+	1417	1314	**1302**	1315	1449

Table 8. Real-world OCR post-correction results as described in text. Different parametrizations and LM weights w_{LM}. Lower `diff` scores are better. In bold: best results in each row.

in recent years and applied them within the spelling error correction paradigm. Moreover, we investigated how a simple 'k-best decoding plus dictionary lookup' strategy performs in this context. We showed that such an approach can significantly outdo baselines such as the edit distance, weighted edit distance, and the noisy channel Brill and Moore model (Brill and Moore, 2000) applied for spelling error correction. In particular, we saw that in the named dictionary-based modus, (three of) the models surveyed here are much better than the baselines in ranking a set of candidate suggestions for a falsely spelled input. We have also shown that by combining the four models surveyed (and the baselines) via simple combination techniques, even better results can be obtained. Finally, we conducted real-world OCR correction experiments based on our trained systems and language models. The data and the dictionaries can be accessed via `https://www.hucompute.org/ressourcen/corpora` so that our findings may be used as a starting point for related research.

In future work, we intend to investigate more sophisticated combination techniques for combining outputs of several spell checkers, e.g., on the character-level, as done in Cortes et al. (2014); Eger (2015d,c); Yao and Kondrak (2015). We also intend to evaluate neural-network based techniques in the present scenario (Sutskever et al., 2014; Yao and Zweig, 2015). Finally, we plan to substitute the CRF++ tagger used in AliSeTra by a higher-order CRF tagger as described by Müller et al. (2013).

Acknowledgments

We gratefully acknowledge financial support by the BMBF via the research project *CompHistSem* (`http://comphistsem.org/home.html`). We also thank Tim Geelhaar and Roland Scheel for providing the OCR scan of the subpart of the Patrologia Latina on which our Latin OCR experiments are based.

Bibliography

Bartlett, Susan, Grzegorz Kondrak, and Colin Cherry. Automatic Syllabification with Structured SVMs for Letter-to-Phoneme Conversion. In McKeown, Kathleen, Johanna D. Moore, Simone Teufel, James Allan, and Sadaoki Furui, editors, *ACL*, pages 568–576. The Association for Computational Linguistics, 2008. ISBN 978-1-932432-04-6. URL `http://dblp.uni-trier.de/db/conf/acl/acl2008.html#BartlettKC08`.

Bisani, Maximilian and Hermann Ney. Joint-sequence models for grapheme-to-phoneme conversion. *Speech Communication*, 50(5):434–451, 2008. URL `http://dblp.uni-trier.de/db/journals/speech/speech50.html#BisaniN08`.

Brill, Eric and Robert C. Moore. An Improved Error Model for Noisy Channel Spelling Correction. In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics (ACL)*, ACL '00, pages 286–293, Stroudsburg, PA, USA, 2000. Association for Computational Linguistics. doi: 10.3115/1075218.1075255. URL `http://dx.doi.org/10.3115/1075218.1075255`.

Cortes, Corinna, Mehryar Mohri, and Jason Weston. A General Regression Technique for Learning Transductions. In *Proceedings of the 22Nd International Conference on Machine Learning*, Proceedings of the International Conference on Machine Learning (ICML), pages 153–160, New York, NY, USA, 2005. ACM. ISBN 1-59593-180-5. doi: 10.1145/1102351.1102371. URL `http://doi.acm.org/10.1145/1102351.1102371`.

Cortes, Corinna, Vitaly Kuznetsov, and Mehryar Mohri. Ensemble Methods for Structured Prediction. In *Proceedings of the 31st International Conference on Machine Learning (ICML)*, 2014.

Cotterell, Ryan, Nanyun Peng, and Jason Eisner. Stochastic Contextual Edit Distance and Probabilistic FSTs. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (ACL)*, Baltimore, June 2014. URL `http://cs.jhu.edu/~jason/papers/#cotterell-peng-eisner-2014`. 6 pages.

Cotterell, Ryan, Nanyun Peng, and Jason Eisner. Modeling Word Forms Using Latent Underlying Morphs and Phonology. *Transactions of the Association for Computational Linguistics*, 3:433–447, 2015. ISSN 2307-387X. URL `https://tacl2013.cs.columbia.edu/ojs/index.php/tacl/article/view/480`.

Cucerzan, S. and E. Brill. Spelling Correction as an Iterative Process that Exploits the Collective Knowledge of Web Users. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2004.

Dreyer, Markus, Jason Smith, and Jason Eisner. Latent-Variable Modeling of String Transductions with Finite-State Methods. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1080–1089. ACL, 2008. URL `http://dblp.uni-trier.de/db/conf/emnlp/emnlp2008.html#DreyerSE08`.

Duan, Huizhong and Bo-June (Paul) Hsu. Online Spelling Correction for Query Completion. In *Proceedings of the 20th International Conference on World Wide Web*, WWW '11, pages 117–126, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0632-4. doi: 10.1145/1963405.1963425. URL `http://doi.acm.org/10.1145/1963405.1963425`.

Eger, Steffen. S-Restricted Monotone Alignments: Algorithm, Search Space, and Applications. In *Proceedings of the Conference on Computational Linguistics (COLING)*, pages 781–798, 2012.

Eger, Steffen. Sequence Segmentation by Enumeration: An Exploration. *Prague Bull. Math. Linguistics*, 100:113–132, 2013. URL `http://dblp.uni-trier.de/db/journals/pbml/pbml100.html#Eger13`.

Eger, Steffen. Designing and comparing G2P-type lemmatizers for a morphology-rich language. In *Fourth International Workshop on Systems and Frameworks for Computational Morphology*, pages 27–40. Springer International Publishing Switzerland, 2015a.

Eger, Steffen. Do we need bigram alignment models? On the effect of alignment quality on transduction accuracy in G2P. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1175–1185, Lisbon, Portugal, September 2015b. Association for Computational Linguistics. URL `http://aclweb.org/anthology/D15-1139`.

Eger, Steffen. Improving G2P from wiktionary and other (web) resources. In *INTERSPEECH 2015, 16th Annual Conference of the International Speech Communication Association, Dresden, Germany, September 6-10, 2015*, pages 3340–3344, 2015c.

Eger, Steffen. Multiple Many-to-Many Sequence Alignment for Combining String-Valued Variables: A G2P Experiment. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 909–919, Beijing, China, July 2015d. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/P15-1088`.

Farra, Noura, Nadi Tomeh, Alla Rozovskaya, and Nizar Habash. Generalized Character-Level Spelling Error Correction. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 161–167, Baltimore, Maryland, June 2014. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/P/P14/P14-2027`.

Gubanov, Sergey, Irina Galinskaya, and Alexey Baytin. Improved Iterative Correction for Distant Spelling Errors. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014, June 22-27, 2014, Baltimore, MD, USA, Volume 2: Short Papers*, pages 168–173, 2014. URL `http://aclweb.org/anthology/P/P14/P14-2028.pdf`.

Gusfield, Dan. *Algorithms on Strings, Trees, and Sequences - Computer Science and Computational Biology*. Cambridge University Press, 1997. ISBN 0-521-58519-8.

Jiampojamarn, Sittichai, Grzegorz Kondrak, and Tarek Sherif. Applying Many-to-Many Alignments and Hidden Markov Models to Letter-to-Phoneme Conversion. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, pages 372–379, Rochester, New York, April 2007. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/N/N07/N07-1047`.

Jiampojamarn, Sittichai, Colin Cherry, and Grzegorz Kondrak. Joint Processing and Discriminative Training for Letter-to-Phoneme Conversion. In *Proceedings of ACL-08: HLT*, pages 905–913, Columbus, Ohio, June 2008. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/P/P08/P08-1103`.

Jiampojamarn, Sittichai, Aditya Bhargava, Qing Dou, Kenneth Dwyer, and Grzegorz Kondrak. DirecTL: a Language Independent Approach to Transliteration. In *Proceedings of the 2009 Named Entities Workshop: Shared Task on Transliteration (NEWS 2009)*, pages 28–31, Suntec, Singapore, August 2009. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/W/W09/W09-3504.

Jiampojamarn, Sittichai, Colin Cherry, and Grzegorz Kondrak. Integrating Joint n-gram Features into a Discriminative Training Framework. In *Proceedings of HLT-NAACL*, pages 697–700. The Association for Computational Linguistics, 2010a. ISBN 978-1-932432-65-7. URL http://dblp.uni-trier.de/db/conf/naacl/naacl2010.html#JiampojamarnCK10.

Jiampojamarn, Sittichai, Kenneth Dwyer, Shane Bergsma, Aditya Bhargava, Qing Dou, Mi-Young Kim, and Grzegorz Kondrak. Transliteration Generation and Mining with Limited Training Resources. In *Proceedings of the 2010 Named Entities Workshop*, pages 39–47, Uppsala, Sweden, July 2010b. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/W10-2405.

Koehn, Philipp. Europarl: A Parallel Corpus for Statistical Machine Translation. In *Conference Proceedings: the tenth Machine Translation Summit*, pages 79–86, Phuket, Thailand, 2005. AAMT, AAMT. URL http://mt-archive.info/MTS-2005-Koehn.pdf.

Kukich, Karen. Techniques for Automatically Correcting Words in Text. *ACM Comput. Surv.*, 24(4):377–439, Dec. 1992. ISSN 0360-0300. doi: 10.1145/146370.146380. URL http://doi.acm.org/10.1145/146370.146380.

Mehler, Alexander, Tim vor der Brück, Rüdiger Gleim, and Tim Geelhaar. Towards a Network Model of the Coreness of Texts: An Experiment in Classifying Latin Texts using the TTLab Latin Tagger. In Biemann, Chris and Alexander Mehler, editors, *Text Mining: From Ontology Learning to Automated Text Processing Applications*, Theory and Applications of Natural Language Processing, pages 87–112. Springer, Berlin/New York, 2015.

Migne, Jacques-Paul, editor. *Patrologiae cursus completus: Series latina*. 1–221. Chadwyck-Healey, Cambridge, 1844–1855.

Mitankin, Petar, Stefan Gerdjikov, and Stoyan Mihov. An Approach to Unsupervised Historical Text Normalisation. In *Proceedings of the First International Conference on Digital Access to Textual Cultural Heritage*, Proceedings of DATeCH '14, pages 29–34, New York, NY, USA, 2014. ACM. ISBN 978-1-4503-2588-2. doi: 10.1145/2595188.2595191. URL http://doi.acm.org/10.1145/2595188.2595191.

Müller, Thomas, Helmut Schmid, and Hinrich Schütze. Efficient Higher-Order CRFs for Morphological Tagging. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 322–332, Seattle, Washington, USA, October 2013. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/D13-1032.

Nicolai, Garrett, Colin Cherry, and Grzegorz Kondrak. Inflection Generation as Discriminative String Transduction. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 922–931, Denver, Colorado, May–June 2015. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/N15-1093.

Novak, Josef Robert, Nobuaki Minematsu, and Keikichi Hirose. Phonetisaurus: Exploring grapheme-to-phoneme conversion with joint n-gram models in the WFST framework. *Natural Language Engineering*, 2015.

Okazaki, Naoaki, Yoshimasa Tsuruoka, Sophia Ananiadou, and Jun'ichi Tsujii. A Discriminative Candidate Generator for String Transformations. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, EMNLP '08, pages 447–456, Stroudsburg, PA, USA, 2008. Association for Computational Linguistics. URL `http://dl.acm.org/citation.cfm?id=1613715.1613772`.

Pirinen, Tommi A. and Krister Lindén. State-of-the-Art in Weighted Finite-State Spell-Checking. In *Computational Linguistics and Intelligent Text Processing - 15th International Conference, CICLing 2014, Kathmandu, Nepal, April 6-12, 2014, Proceedings, Part II*, pages 519–532, 2014. doi: 10.1007/978-3-642-54903-8_43. URL `http://dx.doi.org/10.1007/978-3-642-54903-8_43`.

Raaijmakers, Stephan. A deep graphical model for spelling correction. In *Proceedings BNAIC 2013*, 2013.

Reynolds, L. D. and Nigel Wilson. *Scribes and scholars. A guide to the transmission of Greek and Latin literature*. Clarendon Press, Oxford, 3. aufl. edition, 1991. ISBN 0-19-872145-5.

Rosti, Antti-Veikko I., Necip Fazil Ayan, Bing Xiang, Spyridon Matsoukas, Richard M. Schwartz, and Bonnie J. Dorr. Combining Outputs from Multiple Machine Translation Systems. In Sidner, Candace L., Tanja Schultz, Matthew Stone, and ChengXiang Zhai, editors, *Proceedings of HLT-NAACL*, pages 228–235. The Association for Computational Linguistics, 2007. URL `http://dblp.uni-trier.de/db/conf/naacl/naacl2007.html#RostiAXMSD07`.

Springmann, Uwe, Dietmar Najock, Hermann Morgenroth, Helmut Schmid, Annette Gotscharek, and Florian Fink. OCR of historical printings of Latin texts: problems, prospects, progress. In *Digital Access to Textual Cultural Heritage 2014, DATeCH 2014, Madrid, Spain, May 19-20, 2014*, pages 71–75, 2014. doi: 10.1145/2595188.2595205.

Stolcke, Andreas. SRILM-an extensible language modeling toolkit. In *Proceedings International Conference on Spoken Language Processing*, pages 257–286, November 2002.

Sun, Xu, Jianfeng Gao, Daniel Micol, and Chris Quirk. Learning Phrase-Based Spelling Error Models from Clickthrough Data. In Hajic, Jan, Sandra Carberry, and Stephen Clark, editors, *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 266–274. The Association for Computational Linguistics, 2010. ISBN 978-1-932432-67-1. URL `http://dblp.uni-trier.de/db/conf/acl/acl2010.html#SunGMQ10`.

Sutskever, Ilya, Oriol Vinyals, and Quoc V. Le. Sequence to Sequence Learning with Neural Networks. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 3104–3112, 2014.

Toutanova, Kristina and Robert C. Moore. Pronunciation Modeling for Improved Spelling Correction. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 144–151. ACL, 2002. URL `http://dblp.uni-trier.de/db/conf/acl/acl2002.html#ToutanovaM02`.

vor der Brück, Tim, Alexander Mehler, and Md. Zahurul Islam. ColLex.EN: Automatically Generating and Evaluating a Full-form Lexicon for English. In *Proceedings of LREC 2014*, Reykjavik, Iceland, 2014.

Wang, Ziqi, Gu Xu, Hang Li, and Ming Zhang. A Probabilistic Approach to String Transformation. *IEEE Trans. Knowl. Data Eng.*, 26(5):1063–1075, 2014. doi: 10.1109/TKDE.2013.11. URL http://doi.ieeecomputersociety.org/10.1109/TKDE.2013.11.

Whitelaw, Casey, Ben Hutchinson, Grace Y. Chung, and Gerard Ellis. Using the Web for Language Independent Spellchecking and Autocorrection. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 2 - Volume 2*, EMNLP '09, pages 890–899, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics. ISBN 978-1-932432-62-6. URL http://dl.acm.org/citation.cfm?id=1699571.1699629.

Yao, Kaisheng and Geoffrey Zweig. Sequence-to-Sequence Neural Net Models for Grapheme-to-Phoneme Conversion. *CoRR*, abs/1506.00196, 2015.

Yao, Lei and Grzegorz Kondrak. Joint Generation of Transliterations from Multiple Representations. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 943–952, Denver, Colorado, May–June 2015. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/N15-1095.

Address for correspondence:
Steffen Eger
eger@ukp.informatik.tu-darmstadt.de
Technische Universität Darmstadt
Hochschulstraße 10, 64289 Darmstadt, Germany

The Prague Bulletin of Mathematical Linguistics
NUMBER 105 APRIL 2016 101–110

Gibbs Sampling Segmentation of Parallel Dependency Trees for Tree-Based Machine Translation

David Mareček, Zdeněk Žabokrtský

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

Abstract

We present a work in progress aimed at extracting translation pairs of source and target dependency treelets to be used in a dependency-based machine translation system. We introduce a novel unsupervised method for parallel tree segmentation based on Gibbs sampling. Using the data from a Czech-English parallel treebank, we show that the procedure converges to a dictionary containing reasonably sized treelets; in some cases, the segmentation seems to have interesting linguistic interpretations.

1. Introduction and related work

The context in which words and phrases are translated must be considered in machine translation. There are two basic ways how it is currently done in mainstream statistical machine translation (SMT). First, source-side sequences (phrases) longer than one word are stored together with their target-side equivalents in a "dictionary" (phrase table). Second, a language model rates possible longer sequences on the target side, which – among other things – reduces "boundary friction" between individually translated phrases. In addition, there are discriminative translation models that can profit from various types of features (including those from more distant context) too.

In dependency-tree-based MT, which constitutes the context of our study, the situation is more or less the same. Larger translation units (treelets composed of more than one node) can be used, like in Quirk et al. (2005). Target-side tree models (utilizing the probability of a word conditioned by its parent instead of its left neighbor(s)) can be used too to ensure that chosen target treelets fit together in the tree structure;

 Corresponding author: marecek@ufal.mff.cuni.cz
Cite as: David Mareček, Zdeněk Žabokrtský. Gibbs Sampling Segmentation of Parallel Dependency Trees for Tree-Based Machine Translation. The Prague Bulletin of Mathematical Linguistics No. 105, 2016, pp. 101–110.
doi: 10.1515/pralin-2016-0005.

such a target-language dependency tree model was used in Žabokrtský and Popel (2009) (although the target tree model was combined only with a single-node translation model in this case). Third, the treelet translation model could be discriminative (i.e., capable of using more features from the context) too.

In this paper we focus on extracting a translation dictionary of pairs of source and target treelets from the node-aligned Czech-English parallel treebank CzEng.[1] We segment the trees into smaller parts called treelets. Then we produce a dictionary of (internally aligned) treelet pairs, equipped with source-to-target conditional probabilities (for both language directions) derived from treelet pair counts.[2]

Our approach is novel in two aspects:

- We use Gibbs sampling (Geman and Geman, 1984) for segmenting parallel trees, using a probabilistic model and a set of constraints that limit acceptable treelet pairs.
- We introduce interleaved trees, where nodes on odd levels contain lemmas of content words, whereas nodes on even levels[3] contain compact information on surface morphosyntactic form of the child node that is manifested in the surface sentence form.

The reasons why we use Gibbs sampling instead of exhaustive enumeration of all possible segmentations on both sides are the following. First, this approach leads to a relatively small translation dictionary, since it converges to segmentations that prefer repeated treelets (the rich-get-richer principle). Second, such a sampling approach allows us to describe only what the properties of the desired solutions are (in terms of a probabilistic model in combination with hard constraints on atomic sampling operations), and we do not need any specialized algorithms for finding such solutions – we just run the sampler. This seems to be a big advantage especially in the case of non-isomorphic trees and also because of noise caused by the fully automatic production of CzEng.

In the past, Bayesian methods (such as those based on Gibbs sampling or Pitman-Yor process) have been already used for tree segmentation. The typical purpose was grammar induction, both in constituency and dependency syntax, with Chung et al. (2014) being a representative of the former and Blunsom and Cohn (2010) of the latter. A dictionary of dependency treelet pairs, automatically extracted from parallel dependency trees, was used in the past too (e.g., Quirk et al., 2005; Ding and Palmer, 2004). However, to the best of our knowledge, there is no study merging these two worlds together. We are not aware of any attempt at finding a treelet translation dictionary for the needs of a real MT system using Gibbs sampling.

---

[1] All annotation contained in the treebank results from automatic tools like POS taggers, dependency parsers, and sentence and word aligners, see Bojar et al. (2012).

[2] Using the generated probabilistic treelet translation dictionary in a real MT system is left for further work. Interestingly, it seems that it will be possible to use Gibbs sampling also for decoding.

[3] The technical root added to each sentence is considered the first level.

Unlike the mainstream SMT, our approach relies on a fairly deep level of linguistic abstraction called tectogrammatical trees, as introduced by Sgall (1967), fully implemented for the first time in the Prague Dependency Treebank 2.0 (Hajič et al., 2006), and further adopted for the needs of tree-based MT in the TectoMT translation system (Žabokrtský et al., 2008). Only content words have nodes of their own in tectogrammatical trees, while function words disappear and are possibly turned to attributes inside the tectogrammatical nodes. Nodes of tectogrammatical trees are highly structured (they have tens of attributes, some of which further structured internally). Most of the attributes can be transferred from the source language to the target language relatively easily (for instance, the plural value of the grammatical number attribute goes most often to plural on the target side too). The attributes that are naturally most difficult to translate are *lemma* and *formeme* (the latter specification of the surface form, such as morphological case, or a function word such as a concrete preposition, or a verb clause type, see Dušek et al. (2012)). We follow Mareček et al. (2010) in using machine learning only for translating *lemmas* and *formemes*; the simpler-to-translate attributes are transferred by a less complex by-pass.

Since we want to keep the data structure used in the treelet transfer step as simple as possible, we convert tectogrammatical trees to so called *interleaved trees*, which contain only single-attribute nodes. Each original tectogrammatical node is split into a lemma node and a formeme node as the lemma's parent.[4] Regarding word-alignment, we only adopt the 1-to-1 alignment links from the original data.[5] In the interleaved trees, each such link is split into two: one connecting the *formeme* nodes and the other connecting the *lemma* nodes.

2. Segmentation by sampling

In order to generate a treelet translation dictionary, we need to split the aligned parallel trees from CzEng into smaller parts; we call them *bi-treelets*. Each bi-treelet consists of two subtrees (treelets) of the source and target trees respectively, and of alignment links internally connecting the two subtrees.

Virtually any tree edge can be cut across by the segmentation. However, since the source and the target trees are generally not isomorphic, we define additional constraints in order to receive technically reasonable bi-treelets.

- *Alignment constraint*: A pair of treelets has to be closed under alignment. In other words, no alignment link can refer outside of the bi-treelet.
- *Non-empty constraint*: Each bi-treelet must have at least one node both in the source and in the target tree. This constraint ensures that bi-trees projecting

[4] Valency of a governing word is usually determined by its lexeme (*lemma*), while the requirements imposed on its valency arguments are manifested by morphosyntactic features (*formemes*). Thus it seems more linguistically adequate to place the child's formeme between the parent and child's lemmas.

[5] We employed the links covered by the GIZA++ intersection symmetrization.

some nodes to nothing cannot exist and therefore both source and target dependency trees must be divided into the same number of treelets.

We use the Gibbs sampling algorithm to find the optimal translation bi-treelets. To model the probability of a segmented corpus, we use a generative model based on the Chinese restaurant process (Aldous, 1985). Assume that the corpus C is segmented to n bi-treelets $[B_1, \dots, B_n]$. The probability that such a corpus is generated is

$$P(C) = p_t^{n-1}(1-p_t)\prod_{i=1}^{n} \frac{\alpha P_0(B_i) + \mathrm{count}^{-i}(B_i)}{\alpha + i},$$

where $P_0(B_i)$ is a prior probability of a particular bi-treelet, hyperparameter α determines the strength of the prior, $\mathrm{count}^{-i}(B_i)$ denotes how many times the bi-treelet B_i was generated before the position i, and p_t is the probability of generating the next bi-treelet.

The prior probability of a treelet is computed according to a separate generative micro-story: (1) We generate the node labels from a uniform distribution (probability $1/\#types$) and after each label, we decide whether to continue (probability p_c) or not $(1 - p_c)$, (2) When the labels are generated, we generate the shape of the tree from uniform distribution over all possible dependency trees with k nodes, which is k^{k-1}. This gives us the following formula for the treelet prior probability:

$$P_0(T) = \left(\frac{1}{\#types}\right)^k p_c^{k-1}(1-p_c)\frac{1}{k^{k-1}}$$

The bi-treelet prior probability is then a multiplication of the source and target treelet priors.[6]

Before sampling, we initialize bi-treelets randomly. We assign the binary attribute `is_segmented` to each dependency edge in both source and target trees. Technically, this attribute is assigned to the dependent node. Due to the alignment and non-empty constraints, the following conditions must be met:

- If two nodes are aligned, they must agree in the `is_segmented` attribute. In other words, both the nodes are roots of the bi-treelet or neither of them is.
- If two nodes are aligned, their closest aligned ancestors (parents, grandparents, etc.) should be aligned to each other. If not, there are some crossing alignment links, which could cause disconnected treelets during the sampling. To prevent this, the `is_segmented` attributes of such two nodes are permanently set to 1 and can not be changed during the sampling.
- If a node is not aligned, the `is_segmented` attribute is set permanently to 0 and cannot be changed during the sampling. This property connects all the not-aligned nodes to their closest aligned ancestors and ensure the non-empty constraint.

[6] We do not take into account possibly different alignment of nodes between the treelets.

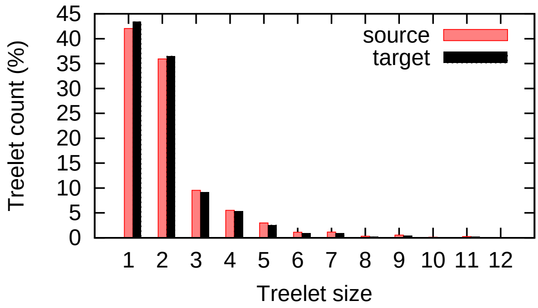


Figure 1. Distribution over different treelet sizes in the dictionary ($\alpha = 0.1$, $T = 1$, $c_p = 0.5$, $c_t = 0.99$).

The sampling algorithm goes through all the nodes in the source trees and samples a new binary value with respect to the corpus probability $P(C)$ (in case the change is not forbidden by the aforementioned constraints). The `is_segmented` attribute of its aligned counterpart in the target tree is set to the same value. Due to the exchangeability property, it is not necessary to compute the whole corpus probability. See the details in Cohn et al. (2009).

After a couple of "burn-in" iterations, the segmentation of trees converges to reasonable-looking bi-treelets. In the remaining iterations, the counts of bi-treelets are collected. Finally, the dictionary of bi-treelets with assigned probabilities computed from collected counts is created.

3. Experiments and evaluation

We perform our experiments on 10% of the Czech-English parallel treebank CzEng 1.0 (Bojar et al., 2012). This subset contains about 1.5 million sentences (21 million Czech tokens and 23 million English tokens) from different sources.

We started with initial setting of hyperparameters $\alpha = 0.1$, $p_c = 0.5$, and $p_t = 0.99$. The algorithm converges quite quickly. After the third iteration, the number of changes in the segmentation is less than 2% per iteration. Therefore we decided to set the "burn-in" period to the first 5 iterations and to start the collecting bi-treelets counts from the sixth iteration. The distribution over different sizes of treelets collected in the dictionary is depicted in Figure 1. There is more than 40% one-node treelets and about 35% two-node treelets. The average number of nodes in the bi-treelet is 2.07 in the source (English) and 1.99 in the target (Czech) side.

It is possible that for the decoding, we will need a dictionary with higher variance (more different treelets), so we use annealing to increase the number of segmentation

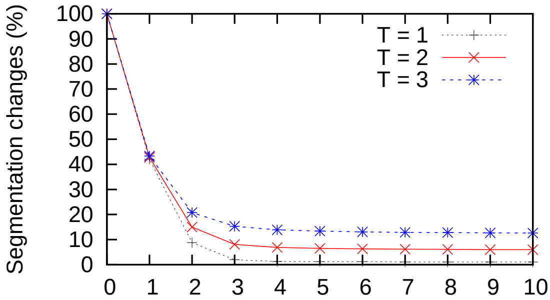


Figure 2. Percentage of changed segmentations during the first 10 iterations for different temperatures ($\alpha = 0.1$, $c_p = 0.5$, $c_t = 0.99$).

Continuation probability p_c	0.5	0.5	0.5	0.5	0.2	0.8	0.8
α	0.001	0.001	0.1	0.1	0.001	0.001	1
Temperature T	1	3	1	3	1	1	2
Last iteration dictionary size	2.45M	2.26M	2.48M	2.32M	2.49M	2.42M	2.34M
Collected dictionary size	2.69M	3.54M	2.73M	3.74M	2.58M	2.78M	3.31M
Average English treelet size	2.19	2.06	2.18	2.05	2.17	2.20	2.16
Average Czech treelet size	2.07	1.96	2.07	1.95	2.06	2.09	2.04

Table 1. The effect of setting the hyperparameters on the dictionary size and other quantities.

changes during the sampling. We introduce a temperature T and exponentiate all the probabilities by $1/T$. Temperatures higher than 1 flatten the distribution and boost the segmentation changes. Figure 2 shows that segmentation changes in the tenth iteration increased to 7% for $T = 2$ and to 12% for $T = 3$.

Table 1 shows the dictionary characteristics for different parameter settings. As expected, the collected dictionary size grows with growing temperatures, while the size of the dictionary based on the last iteration slightly decreases. Therefore, it will be easy to control the trade-off between the size of generated dictionary and the sharper distribution of translation candidates. Different values of the hyperparameter α do not affect the results much. Similarly, the continuation probability p_c does not affect the sizes of bi-treelets much.

We inspected the segmented trees after the last iteration; an example is shown in Figure 3. The thin edges are the ones cut by the segmentation, and the thick edges represent the delimited treelets (there are four bi-treelets in the figure). The lemma node and its respective formeme node often belong to the same treelet. Collocations

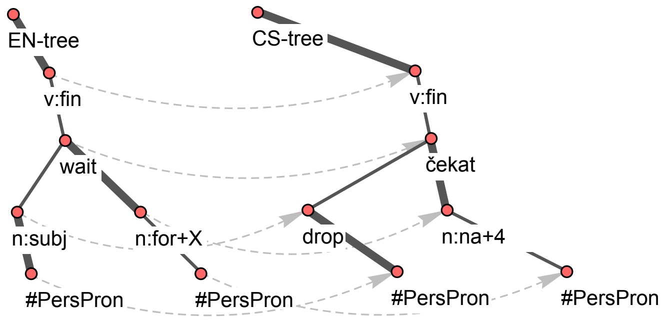


Figure 3. Interleaved trees representing the sentences "Čekal jsem na tebe." – "I've been waiting for you." and their segmentation to bi-treelets.

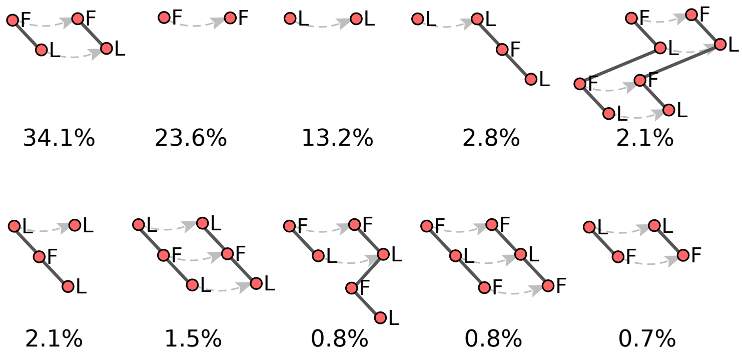


Figure 4. Distribution of the most frequent bi-treelets types in the dictionary (L = lemma, F = formeme).

(e.g. "European Union") also tend to constitute treelets of their own. The observation which we find most interesting is the manifestation of parallel verb valency captured by some treelets, such as the aligned formeme nodes *n:for+X – n:na+4* that are stuck to their governing verbs *wait – čekat* in a bi-treelet and not to their children.

Figure 4 shows 10 most frequent types of bi-treelets. We can see that if a pair of *formeme* nodes is inside a larger treelet it is connected to its respective pair of *lemma* nodes. Exceptions are the last two types of bi-treelets, where the *formeme* nodes are leaves. These are the cases of stronger valency between a parent *lemma* and morphosyntactic form of its dependent (e.g. *wait* + *n:for+X*).

4. Conclusions

We show a new method for obtaining a treelet-to-treelet translation dictionary from a parallel treebank using Gibbs sampling. In future work, we will evaluate our approach in a tree-based MT system.

Acknowledgments

The work was supported by the grant 14-06548P of the Czech Science Foundation. It has been using language resources developed, stored and distributed by the LINDAT/CLARIN project of the Ministry of Education, Youth and Sports of the Czech Republic (project LM2010013).

Bibliography

Aldous, David. Exchangeability and related topics. In *Ecole d'Ete de Probabilities de Saint-Flour XIII 1983*, pages 1–198. Springer, 1985.

Blunsom, Phil and Trevor Cohn. Unsupervised Induction of Tree Substitution Grammars for Dependency Parsing. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, EMNLP '10, pages 1204–1213, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics. URL `http://dl.acm.org/citation.cfm?id=1870658.1870775`.

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 8th International Conference on Language Resources and Evaluation (LREC 2012)*, pages 3921–3928, İstanbul, Turkey, 2012. European Language Resources Association. ISBN 978-2-9517408-7-7.

Chung, Tagyoung, Licheng Fang, Daniel Gildea, and Daniel Stefankovic. Sampling Tree Fragments from Forests. *Computational Linguistics*, 40(1):203–229, 2014. doi: 10.1162/COLI_a_00170. URL `http://dx.doi.org/10.1162/COLI_a_00170`.

Cohn, Trevor, Sharon Goldwater, and Phil Blunsom. Inducing compact but accurate tree-substitution grammars. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 548–556, Boulder, Colorado, 2009.

Ding, Yuan and Martha Stone Palmer. Automatic Learning of Parallel Dependency Treelet Pairs. In Su, Keh-Yih, Jun ichi Tsujii, Jong-Hyeok Lee, and Oi Yee Kwong, editors, *IJCNLP*, volume 3248 of *Lecture Notes in Computer Science*, pages 233–243. Springer, 2004. ISBN 3-540-24475-1.

Dušek, Ondřej, Zdeněk Žabokrtský, Martin Popel, Martin Majliš, Michal Novák, and David Mareček. Formemes in English-Czech Deep Syntactic MT. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*, pages 267–274, Montréal, Canada, 2012. Association for Computational Linguistics. ISBN 978-1-937284-20-6.

Geman, Stuart and Donald Geman. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6:721–741, 1984.

Hajič, Jan, Jarmila Panevová, Eva Hajičová, Petr Sgall, Petr Pajas, Jan Štěpánek, Jiří Havelka, Marie Mikulová, Zdeněk Žabokrtský, and Magda Ševčíková-Razímová. Prague Dependency Treebank 2.0. CD-ROM, Linguistic Data Consortium, LDC Catalog No.: LDC2006T01, Philadelphia, 2006.

Mareček, David, Martin Popel, and Zdeněk Žabokrtský. Maximum Entropy Translation Model in Dependency-Based MT Framework. In *Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and MetricsMATR*, pages 201–201, Uppsala, Sweden, 2010. Uppsala Universitet, Association for Computational Linguistics. ISBN 978-1-932432-71-8.

Quirk, Chris, Arul Menezes, and Colin Cherry. Dependency Treelet Translation: Syntactically Informed Phrasal SMT. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, ACL '05, pages 271–279, Stroudsburg, PA, USA, 2005. Association for

Computational Linguistics. doi: 10.3115/1219840.1219874. URL http://dx.doi.org/10.3115/1219840.1219874.

Sgall, Petr. *Generativní popis jazyka a česká deklinace*. Academia, Praha, 1967.

Žabokrtský, Zdeněk and Martin Popel. Hidden Markov Tree Model in Dependency-based Machine Translation. In *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*, pages 145–148, Suntec, Singapore, 2009. Association for Computational Linguistics. ISBN 978-1-932432-61-9.

Žabokrtský, Zdeněk, Jan Ptáček, and Petr Pajas. TectoMT: Highly Modular MT System with Tectogrammatics Used as Transfer Layer. In *ACL 2008 WMT: Proceedings of the Third Workshop on Statistical Machine Translation*, pages 167–170, Columbus, OH, USA, 2008. Association for Computational Linguistics. ISBN 978-1-932432-09-1.

Address for correspondence:
David Mareček
marecek@ufal.mff.cuni.cz
Institute of Formal and Applied Linguistics
Faculty of Mathematics and Physics,
Charles University in Prague
Malostranské náměstí 25
118 00 Praha 1, Czech Republic

PBML

The Prague Bulletin of Mathematical Linguistics

NUMBER 105 APRIL 2016 111–142

A Minimally Supervised Approach for Synonym Extraction with Word Embeddings

Artuur Leeuwenberg[a], Mihaela Vela[b], Jon Dehdari[bc], Josef van Genabith[bc]

[a] KU Leuven – University of Leuven, Belgium
[b] Saarland University, Germany
[c] DFKI, German Research Center for Artificial Intelligence, Germany

Abstract

In this paper we present a novel approach to minimally supervised synonym extraction. The approach is based on the word embeddings and aims at presenting a method for synonym extraction that is extensible to various languages.

We report experiments with word vectors trained by using both the continuous bag-of-words model (CBoW) and the skip-gram model (SG) investigating the effects of different settings with respect to the contextual window size, the number of dimensions and the type of word vectors. We analyze the word categories that are (cosine) similar in the vector space, showing that cosine similarity on its own is a bad indicator to determine if two words are synonymous. In this context, we propose a new measure, relative cosine similarity, for calculating similarity relative to other cosine-similar words in the corpus. We show that calculating similarity relative to other words boosts the precision of the extraction. We also experiment with combining similarity scores from differently-trained vectors and explore the advantages of using a part-of-speech tagger as a way of introducing some light supervision, thus aiding extraction.

We perform both intrinsic and extrinsic evaluation on our final system: intrinsic evaluation is carried out manually by two human evaluators and we use the output of our system in a machine translation task for extrinsic evaluation, showing that the extracted synonyms improve the evaluation metric.

 Corresp. author: tuur.leeuwenberg@cs.kuleuven.be
Cite as: Artuur Leeuwenberg, Mihaela Vela, Jon Dehdari, Josef van Genabith. A Minimally Supervised Approach for Synonym Extraction with Word Embeddings. The Prague Bulletin of Mathematical Linguistics No. 105, 2016, pp. 111–142. doi: 10.1515/pralin-2016-0006.

1. Introduction

The research presented here explores different methods to extract synonyms from text. We try to do this using as little supervision as possible, with the goal that the method can be applied to multiple languages.

1.1. Motivation

The initial motivation for our research comes from machine translation (MT) evaluation. MT output to be evaluated is referred to as a *hypothesis translation*. A *reference translation* is a translation produced by a proficient human translator. To evaluate an MT system, hypothesis translations are compared with reference translations. This comparison is often done automatically.

While simple automatic evaluation approaches (Snover et al., 2006; Papineni et al., 2002; Doddington, 2002) are based on exact (sub-)string matches between hypotheses and references, more recent evaluation methods are using machine learning approaches (Stanojević and Sima'an, 2014; Gupta et al., 2015b; Vela and Tan, 2015; Vela and Lapshinova-Koltunski, 2015) to determine the quality of machine translation. More sophisticated approaches such as Meteor (Denkowski and Lavie, 2014; Banerjee and Lavie, 2005), Asiya (Gonzà lez et al., 2014), and VERTa (Comelles and Atserias, 2014), incorporate lexical, syntactic and semantic information into their scores, attempting to capture synonyms and paraphrases, to better account for hypotheses and references that differ in form but are similar in meaning.

Meteor computes an alignment between the hypothesis and reference to determine to what extent they convey the same meaning. Alignments are defined by what parts of the two sentences can match. Finding possible matches is done by means of four modules (1) exact matching, (2) stemmed matching, (3) synonym matching, and (4) paraphrase matching. Exact matching uses string identity between tokens, stemmed matching between stemmed tokens. Paraphrase matching employs a paraphrase database to match phrases which may not be string identical. The synonym module does the same for words and uses a synonym database resource. For example, the best alignment for the hypothesis sentence 1 and the reference sentence 2 is shown in Figure 1.

(1) *Hypothesis:*
The practiced reviewer chose to go through it consistently.

(2) *Reference:*
The expert reviewers chose to go through it in a coherent manner.

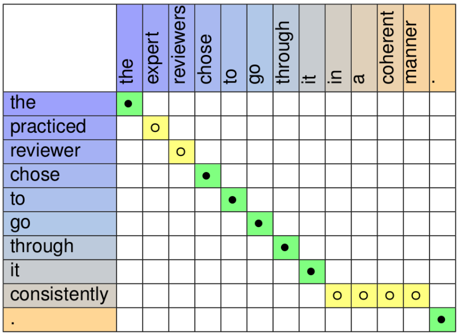


Figure 1. Meteor 1.5 alignment of hypothesis sentence 1, and reference sentence 2

In Figure 1, exact matches are indicated by black dots. The stemming module matched "reviewer" with "reviewers". The paraphrase module matched "consistently" with "in a coherent manner", and the synonym module matched "practiced" with "expert".

Three of these matching modules use language-dependent resources. Paraphrases and synonyms come from a pre-constructed lexical database, and stemming happens with a pre-trained stemmer. For this reason, not all modules are available for all languages. Currently in Meteor 1.5, the synonym module is only available for English. The module uses synonyms from the lexical database WordNet (Miller, 1995). Manual construction of lexical resources such as WordNet is time consuming and expensive, and needs to be done for each different language.

By contrast, large text resources are available for many languages. In our research we investigate whether, and if so to what extent, it is possible to automatically extract synonym resources from raw text using unsupervised or minimally supervised methods based on the *distributional hypothesis*: words that occur in the same contexts tend to have similar meanings (Harris, 1954). In particular we use word embeddings, i.e. dense distributional word vectors (Mikolov et al., 2013a), to compute similarity between words. We develop a new similarity metric, relative cosine similarity, and show that this metric improves the extraction of synonyms from raw text. We evaluate our method using both intrinsic and extrinsic evaluation: we use human evaluation to judge the quality of synonyms extracted and employ the extracted synonyms in the synonymy module of Meteor.

1.2. Word and Synonym

In most recent works on synonym extraction the synonyms from WordNet are used for evaluation. In WordNet, synonyms are described as "words that denote the same concept and are interchangeable in many contexts". In the current work, our notion of words is merely a string of characters. Since there is *homography*, i.e. one word can have different lemmas, with different meanings and origins, we modifiy this notion of synonyms slightly. We think of *synonyms* as words that denote the same concept and are interchangeable in many contexts, with regard to one of their senses.

1.3. Outline

In Section 2, we will proceed to describe the distributional word vectors we used in our experiments, and the related work in synonym extraction. In Section 3 we describe different experiments in which we explore synonym extraction using the continuous bag-of-words model and the skip-gram model. Section 4 describes and evaluates a few methods that introduce some supervision, such as using a part-of-speech tagger. In Section 5 we do an evaluation of a system that combines different proposed findings, for English and German. We evaluate manually, and additionally by using the extracted synonyms for the task of machine translation evaluation. Section 6 concludes the article by giving a summary of the findings and possibilities for future work.

2. Related Work

2.1. Distributional Word Vectors

Distributional word vectors, or *word embeddings*, are word representations that can be constructed from raw text, or a collection of documents, based on their context. The representation of each word will be a vector of numbers, usually real numbers. In some cases linguistic information, such as word dependency information, or morphological information, is also used during the construction process (Levy and Goldberg, 2014; Luong et al., 2013). These word vector representations can then be used to calculate, for example, word similarity and have a wide application domain.

In the last few years many new methods have been proposed to construct distributional word vectors based purely on raw text (Mikolov et al., 2013a; Pennington et al., 2014, *inter alia*). Some methods also use the document structure that can be present in the data (Huang et al., 2012; Liu et al., 2015a,b).

In this work, we experiment mostly with word vectors trained using the *continuous bag-of-words model* (CBoW), and the *skip-gram model* (SG) developed by Mikolov et al. (2013a). It has been shown that these vectors, especially the skip-gram model, can also encode relations between words in a consistent way (Mikolov et al., 2013b). This means that they not only encode word similarity, but also similarity between pairs of words. For example, the offset between the vectors for "queen" and "king" lies very close to the offset between "woman" and "man", i.e. $v(queen) - v(king) \approx v(woman) - v(man)$.

This property has been exploited to extract hypernyms from raw text by Fu et al. (2014) and Tan et al. (2015). The work of Fu et al. (2014) automatically learned, in a supervised way, a piece-wise linear projection that can map a word to its hypernym in the word vector space, for Chinese. To do this they clustered the vector offsets ($v_1 - v_2$), and then found a projection for each cluster. Using this method they could successfully find hypernym pairs. Tan et al. (2015) searched for hypernym pairs in English. They also projected a word to its hypernym in the word vector space. However, instead of automatically learning this projection by using a thesaurus, they concatenated the words "is", and "a" into an "is_a" token in the corpus, and used this as projection. So, $v(w) + v(is\_a)$ would lie very close to the vector for the hypernym of word w.

Both the CBoW and the SG model can be seen as a simplified feedforward neural network, that is constructed from a word and its context. The architecture of the network is shown in Figure 2. CBoW word representations are optimized for predicting the word from its context, the surrounding words. SG word representations are optimized for predicting the context from the word, i.e. given the word, predicting its surrounding words.

In Figure 2, the word is represented as $w(t)$; the contextual window, here of size 2 (two words to the left, and two to the right), is represented as $w(t-2), w(t-1), w(t+1)$, and $w(t+2)$. The final word vector is built from the weights of the projection layer. During training, the window iterates over the text, and updates the weights of the network. Two training methods were described by Mikolov et al. (2013a), namely *hierarchical softmax*, and *negative sampling*. In (hierarchical) softmax, the weights are updated based on the maximization of log-likelihood. In negative sampling, the weights get updated based on whether or not the target word is drawn from the training set, or from a random distribution. The implementation in word2vec[1] has been shown to be quite fast for training state-of-the-art word vectors.

[1] https://code.google.com/p/word2vec/

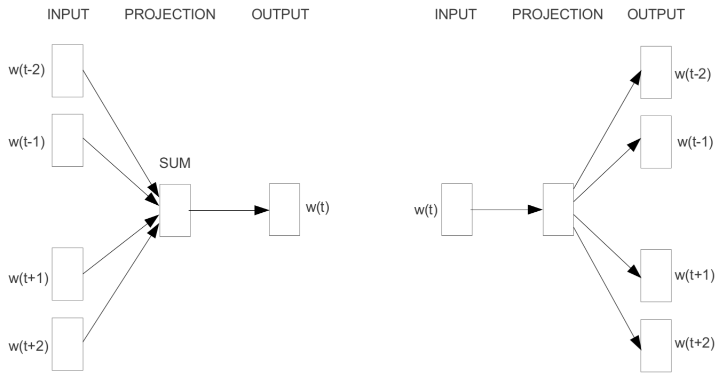


Figure 2. Continuous bag-of-words architecture on the left, and skip-gram on the right.

Depending on the application, it can be beneficial to modify pre-trained word vectors towards specific properties. Faruqui et al. (2015) refined a vector space using relational information, such synonymy and hypernymy, from a lexical database. For the task of antonym detection, Ono et al. (2015) transformed a pre-trained vector space by minimizing the similarity between synonyms and maximizing the similarity between antonyms. Since we would like to use as little supervision as possible, we did not resort to these particular methods.

2.2. Synonym Extraction

Many methods that have been developed for synonym extraction use three main ideas. Firstly, the distributional hypothesis (Van der Plas and Tiedemann, 2006; Agirre et al., 2009; Gupta et al., 2015a; Saveski and Trajkovski, 2010; Pak et al., 2015; Plas and Bouma, 2005). Secondly, the assumption that words that translate to the same word have the same, or a very similar, meaning (Van der Plas and Tiedemann, 2006; Gupta et al., 2015a; Saveski and Trajkovski, 2010; Lin et al., 2003) . And third, the use of linguistic patterns that are typical, or atypical for synonyms to occur in (Lin et al., 2003; Yu et al., 2002).

Van der Plas and Tiedemann (2006) used both distributional word similarity, and translational context for synonym extraction in Dutch. They used a large monolingual corpus to construct a measure for distributional similarity, which was based on grammatical relations. Furthermore, they used different parallel corpora, and automatic alignment, for the construction of a translational context. A contexual similarity measure is constructed to rank the best synonym candidates. The authors remark that when only using distributional similarity there were some word categories that show up frequently but are not synonyms, but rather antonyms, (co)hyponyms, or hypernyms. When using the translational context, these error categories were less frequent, and more synonyms were found. In 2010, an adaptation of the method achieved 31.71% precision at the best candidate (P@1) for high frequency words (most frequent $\frac{1}{3}$ of the vocabulary), 16.22% for low frequency words (least frequent $\frac{1}{3}$), and 29.26% for remaining middle frequency words (van der Plas et al., 2010). Evaluation was done using a selection of 3000 words from Dutch EuroWordNet (Vossen, 1998).

It is very difficult to compare different methods of synonym extraction by only looking at their performance measures, as most papers use different ways to evaluate their approach. They use different word frequency ranges, language(s), textual resources, and gold standard synonyms. These can all have a large influence on the final evaluation.

The word categories mentioned by Van der Plas and Tiedemann (2006) seem to be a common problem when using purely distributional methods (Pak et al., 2015; Plas and Bouma, 2005; Lin et al., 2003). However, the advantage of using methods based on distributional properties is that the coverage is usually greater than that of manually constructed corpora, as Lin et al. (2003) also observed. They tackle the problem of discriminating synonyms from other strongly related words using linguistic patterns. They mention some English patterns in which synonyms hardly occur, like “from X to Y”, and “either X or Y”.

Rather than filtering by means of linguistic patterns, Yu et al. (2002) used particular patterns in which synonyms occur frequently. Their application domain was finding synonyms for gene and protein names. They found that in MEDLINE abstracts synonyms are often listed by a slash or comma symbol. This is probably a more domain dependent pattern. Some other patterns they found were “also called”, or “known as”, and “also known as”.

In this work, we do not resort to a pattern based approach, as they are language and domain dependent.

3. Synonyms in Word Vector Space

In this Section we explain different experiments we carried out to analyze how synonyms behave in different word vector spaces. First, we analyze the effect of contextual window size, the number of dimensions, and the type of word vectors on the precision of extraction, for English and German. Secondly, we look closely at the word categories that are (cosine) similar in the vector space. Then, we look at cosine similarity and introduce relative cosine similarity. Lastly, we examine the overlap of the most similar words in different vector spaces.

3.1. Data and Preprocessing

For English and German we use a 150 million word subset of the NewsCrawl corpus from the 2015 Workshop on Machine Translation[2]. As preprocessing for both languages, we apply lowercasing, tokenization, and digit conflation. In this work, we do not deal with multiword units. For example, for a separable verb in German or English (e.g. abholen / to pick up) can only be found as one word in infinitival or past perfect form (abgeholt/picked up).

We only consider the vocabulary of words that occur at least 10 times in the corpus to ensure that the vectors have a minimum quality. We randomly split the vocabulary into a training, development, and testing set with proportions 8:1:1 respectively. We used vocabularies $S_{\mathtt{train}}$, and $S_{\mathtt{dev}}$ in the experiments to explore, and analyze the different methods described in the paper. After all initial experiments were done, we ran the experiments again using $S_{\mathtt{test}}$ instead of $S_{\mathtt{dev}}$ to evaluate our method. In Table 1, statistics about these vocabularies are given.

Language	Corpus	V	$V_{\geq 10}$	$S_{V_{\geq 10}}$	V_{train}	S_{train}	V_{dev}	S_{dev}	V_{test}	S_{test}
English	150M	650.535	136.821	21.098	109.454	16.882	13.681	2.116	13.683	2.100
German	150M	2.421.840	279.325	16.304	223.458	13.056	27.933	1.599	27.933	1.649

Table 1. Dataset Statistics: V indicates the size of the full corpus vocabulary, $V_{\geq 10}$ indicates the vocabulary size for words with counts greater than or equal to 10. S_x indicates the number of words for which at least one synonym is known, that also occurs in $V_{\geq 10}$.

For evaluation, we use the synonyms from WordNet 3.0 for English, and GermaNet 10.0 for German. In both WordNet and GermaNet words carry a corresponding part-of-speech. In WordNet these are nouns, verbs, adjectives, and adverbs. In GermaNet, synonyms are given for nouns, verbs, and adjectives. Because a given word's part of speech is unknown here, we consider the

[2]http://www.statmt.org/wmt15/translation-task.html

synonyms of each word to be those of all the parts of speech it can potentially have in WordNet or GermaNet.

3.2. Evaluation

We evaluate several experiments in terms of precision, recall and f-measure. *Precision* (P) is calculated as the proportion of correctly predicted synonym word pairs from all predictions. Because synonymy is symmetric, we consider the word pair (w_1, w_2) equivalent to (w_2, w_1) during evaluation. *Recall* (R) is calculated as the proportion of synonym pairs that were correctly predicted from all synonym pairs present in WordNet, or GermaNet. In the experiments we sometimes only search for synonyms of words from a subset of the vocabulary (S_{train} or S_{test}). In this case, recall is calculated only with regard to the synonym pairs from WordNet or GermaNet that involve a word from the mentioned subset. *F-measure* is given by:

$$F = 2 \cdot \frac{P \cdot R}{P + R}$$

3.3. Quantitative Analysis of Training Parameters

In this experiment, we trained CBoW, SG, and *Global Vectors* (GloVe) (Pennington et al., 2014) with different training parameters, and evaluated synonym precision for the $\{1^{st}, 2^{nd}, 4^{th}\}$-most-similar word(s), for vocabulary S_{train}. With similarity we refer to cosine similarity. The hyperparameters we varied are the contextual window size, and the number of dimensions of the vectors. The window size varied over {2, 4, 8, 16, 32}. The number of dimensions varied over {150, 300, 600, 1200}. The experiment is conducted for both English and German, and used 150M training tokens per language. We fixed the number of training iterations: 5 for CBoW and SG, and 25 for GloVe. For CBoW and SG training we used negative sampling with 5 negative samples [3].

The results for the CBoW and SG vectors, for both English and German, are shown in Tables 2, 3, 4, and 5. We excluded the results for the GloVe vectors, as they showed lower precision than SG and CBOW, and we did not use them in further experiments. The general trends of the GloVe vectors were that they had higher precision for larger window sizes. The vectors with highest precision of 0.067 for English were of dimension 300, with a window size of 32. For German, the highest precision was 0.055, and the vectors were of dimension 1200, with a window size of 32 as well.

[3] These are the default values given by the respective authors.

English CBoW																				
dim.	150					300					600					1200				
win.	2	4	8	16	32	2	4	8	16	32	2	4	8	16	32	2	4	8	16	32
P-1	0.077	0.076	0.072	0.066	0.058	0.084	0.083	0.079	0.072	0.068	0.086	**0.086***	0.081	0.074	0.068	0.083	0.083	0.082	0.073	0.067
P-2	0.058	0.056	0.055	0.051	0.046	0.062	0.061	0.059	0.055	0.052	0.063	0.063	0.060	0.056	0.052	0.061	0.061	0.060	0.055	0.050
P-4	0.039	0.039	0.038	0.036	0.032	0.042	0.042	0.041	0.039	0.036	0.043	0.043	0.042	0.039	0.036	0.042	0.042	0.041	0.039	0.036

*Table 2. Precision for different window sizes and number of dimensions, using the **CBoW** model, for **English**.*

English Skip-gram																				
dim.	150					300					600					1200				
win.	2	4	8	16	32	2	4	8	16	32	2	4	8	16	32	2	4	8	16	32
P-1	0.069	0.062	0.055	0.048	0.044	0.069	0.062	0.053	0.048	0.044	0.066	0.059	0.046	0.043	0.039	0.061	0.051	0.039	0.034	0.030
P-2	0.050	0.045	0.040	0.037	0.034	0.050	0.046	0.039	0.036	0.033	0.049	0.044	0.035	0.032	0.030	0.045	0.039	0.029	0.026	0.024
P-4	0.034	0.032	0.028	0.026	0.024	0.034	0.032	0.028	0.025	0.024	0.033	0.030	0.025	0.023	0.021	0.031	0.026	0.020	0.018	0.017

*Table 3. Precision for different window sizes and number of dimensions, using the **Skip-gram** model, for **English**.*

German CBoW																				
dim.	150					300					600					1200				
win.	2	4	8	16	32	2	4	8	16	32	2	4	8	16	32	2	4	8	16	32
P-1	0.073	0.082	0.082	0.083	0.080	0.076	0.084	0.086	0.086	0.082	0.076	0.087	**0.089***	0.088	0.080	0.076	0.083	0.086	0.085	0.081
P-2	0.052	0.057	0.057	0.058	0.056	0.054	0.060	0.062	0.061	0.059	0.054	0.060	0.062	0.062	0.059	0.053	0.059	0.062	0.060	0.058
P-4	0.034	0.036	0.038	0.038	0.037	0.036	0.039	0.041	0.040	0.039	0.035	0.039	0.041	0.041	0.040	0.035	0.039	0.041	0.040	0.039

*Table 4. Precision for different window sizes and number of dimensions, using the **CBoW** model, for **German**.*

German Skip-gram																				
dim.	150					300					600					1200				
win.	2	4	8	16	32	2	4	8	16	32	2	4	8	16	32	2	4	8	16	32
P-1	0.065	0.068	0.066	0.064	0.064	0.064	0.069	0.064	0.062	0.060	0.063	0.064	0.057	0.051	0.049	0.061	0.059	0.046	0.039	0.035
P-2	0.048	0.049	0.049	0.046	0.046	0.048	0.049	0.048	0.045	0.046	0.047	0.046	0.042	0.039	0.037	0.046	0.043	0.035	0.030	0.027
P-4	0.032	0.033	0.032	0.032	0.031	0.033	0.033	0.032	0.031	0.031	0.031	0.031	0.029	0.027	0.026	0.031	0.029	0.025	0.022	0.020

*Table 5. Precision for different window sizes and number of dimensions, using the **Skip-gram** model, for **German**.*

In general, it can be noticed from Tables 2, 3, 4, and 5 that the CBoW vectors give higher precision than SG for both German and English. A reason for this could be that CBoW vectors tend to be slightly more syntactical compared to SG vectors. It could be that the syntactical constraint on synonyms, as they are to appear in similar contexts, has enough influence for CBoW vectors to perform better.

It can also be noticed that for English, smaller contextual windows (2 and 4) generally give better precision, for both CBoW and SG vectors. For German, the optimal window size lies between 8 and 16 for CBoW, and around 4 for SG vectors. The difference in optimal window sizes between English and German could be due to the difference in types of synonyms that are available. WordNet contains synonyms for nouns, verbs, adjectives and adverbs, whereas GermaNet does not include synonyms for adverbs. It could be that adverbs require only a small contextual window to be predicted, compared to nouns, verbs, and adjectives. Another observation that can be made is that for both English and German the optimal window size for SG tends to be slightly lower than for CBoW vectors. Again, this can be due to training difficulty. A larger window can make the training of the SG model more difficult, as a bigger context is to be predicted from one word.

To get an impression of the performance if we would use the most-similar words as synonyms, we calculated precision, recall and f-measure on the test set S_{test}. For English, using the CBoW vectors of dimension 600 with window size 4, precision is 0.11, recall 0.03, and f-measure is 0.05. For German, using a CBoW model of dimension 600 with a window size of 8, precision is 0.08, recall is 0.05, and f-measure 0.06. For both languages these scores are very low. In the next section, we look at some frequent error categories, with the goal to get more insight into the reason behind these low scores.

3.4. Distributionally Similar Words

Only looking at precision, calculated on WordNet or GermaNet, allows us to compare different vector spaces with regard to finding synonyms. However, it might not reflect actual precision, due to lack of coverage of WordNet and GermaNet. Also, it gives only few cues for possible improvements.

For this reason, we also looked more in depth at the most similar words. For 150 randomly chosen English words from S_{train} we looked at the most-similar word, as well as the 2nd-most-similar words, and categorized them. This was done manually. Categories were made based on what was found during the analysis. The word vectors used to create the most similar and 2nd-most-similar words were from the CBoW model of dimension 600, with window size 2, from the previous experiment. The results from this analysis are shown in Table 6. The categories we found are the following:

- *WordNet-Synonyms*: Synonyms as given in WordNet.
- *Human-judged Synonyms*: Synonyms judged by a fluent, but non-native, English speaker.
- *Spelling Variants*: Abbreviations, differences between American and British spelling, and differences in hyphenations.
- *Related:* The two words are clearly semantically related, but not consistently enough to make a separate category.
- *Unrelated / Unknown*: The relation between the two words is unknown.
- *Names*: Names of individuals, groups, institutions, cities, countries or other topographical areas.
- *Co-Hyponyms*: The two words share a close hypernym.
- *Inflections / Derivations*: Inflections or derivations other than plural.
- *Plural*: The found word is the plural version of the given word.
- *Frequent collocations*: The two words occur frequently next to each other.
- *Hyponyms*: The found word is conceptually more specific.
- *Contrastive*: There is an opposition or large contrast between the meaning of the two words.
- *Hypernym*: The found word is conceptually more general.
- *Foreign*: A non-English word.

What can be noticed from Table 6 is that the number of human-judged synonyms is about twice as large as the number of synonyms given by WordNet, even though WordNet considers spelling variants also to be synonyms. This suggests that the actual precision may lie a corresponding amount higher. Where WordNet would give a precision of 0.12 for this set of words, the human annotation gives 0.25. A reason for this large difference can be that resources like WordNet are usually constructed by manually adding the synonyms for a given word. This requires the annotator to think of all the word senses of a word, and their synonyms. This can be a difficult task. Here, the two words are presented and the question is whether they are synonyms. It is probably easier to find the corresponding word senses of both words in this case.

The two biggest error categories are the related words, and unknowns. Since both categories are rather vaguely defined, and consisting of many subcategories we will not go into much more detail on these. There appears some overlap with the error types that were also found by Lin et al. (2003), Plas and Bouma (2005) and Pak et al. (2015), namely co-hyponyms, and hyponyms. However, contrastives and hypernyms are not as frequent in our experiment. Some other major error categories we found are different types of inflections

Category	1st-most-similar	2nd-most-similar	Example
WordNet-Synonyms	18	7	laundry / washing
Human-Synonyms	29	20	masking / obscuring
Spelling Variants	8	4	commander / cmdr
Related	27	33	head-on / three-vehicle
Unrelated/Unknown	13	20	gat / por
Names	15	15	consort / margherete
Co-hyponyms	15	13	sunday / saturday
Inflections/Derivations	12	10	figuring / figured
Plural	11	2	tension / tensions
Frequent Collocations	7	5	dragon / lantern
Hyponyms	5	12	swimsuit / bikini
Contrastive	3	7	rambunctious / well-behaved
Hypernym	2	4	laundry / chores
Foreign	2	4	inhumation / éventualité

Table 6. Counts per category for the most similar word and second most similar word, of 150 randomly chosen English words, in a CBoW model of dimension 600 with a window size of 2.

and derivations, and in particular plurals. This category is not a major problem for our application—machine translation evaluation—as the inflections might already have been matched by the stem module of Meteor. Another category that is fairly frequent involves names. The reason is probably that names might not have many single-word synonyms. The error category of frequent collocations can be explained by the fact that both words usually occur together, and are thus trained on a set of very similar contexts.

3.5. Relative Cosine Similarity

One idea we tested with the goal of improving precision was to only consider word pairs that have very high cosine similarity. In practice this would mean setting a threshold, and only consider those word pairs that have a cosine similarity higher than the threshold. Our expectation was that synonyms are most similar compared to the other word relations. We plotted precision, recall and f-measure on S_{train} against the cosine similarity threshold. This is shown in Figure 3.

What we found however, is that even increasing the cosine similarity threshold does not give an increase in precision. It does not even reach the precision we achieved from our baseline of taking the most-similar word. This indicates that cosine similarity on its own is not a good indicator for synonymy. Still, we get higher precision with choosing the most-similar word. We man-

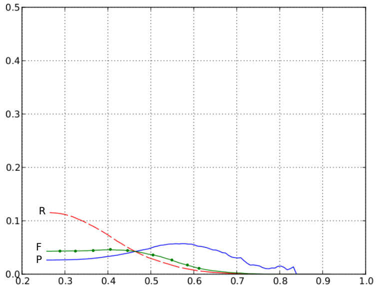


Figure 3. Precision, recall, and f-measure on $S_{\mathtt{train}}$ plotted against the cosine similarity threshold.

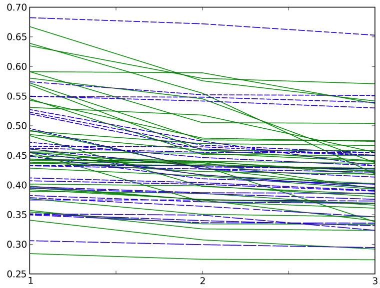


Figure 4. Cosine similarity against n-most similar position, for the 3-most-similar words, where the most-similar word is a synonym or a related word (dashed).

ually looked at the top 10 most-similar words of the 150 words from the previous section, and their cosine similarity. We noticed that when a synonym, inflection or contrastive occurs in the top 10, their cosine similarity is usually much higher than that of the other words in the top 10. That is, the difference in cosine-similarity between the most-similar word, and the second-most-similar word is very high for these categories. When we looked at this for other categories such as co-hyponyms, unknowns, and simply related words, this was not the case. This can be seen when we plot the cosine similarity of the 3-most-similar words for synonyms, and related words taken from the previous experiment.

This is plotted in Figure 4, from which two things can be noticed. Firstly, it is hardly possible to separate the start, at position 1, of the solid lines (synonyms) from the dashed lines (related words) by means of a horizontal cosine threshold. This corresponds to the observation we made earlier, that a cosine similarity threshold does not increase precision. Secondly, many solid lines tend to decrease, and many dashed lines stay relatively horizontal. This indicates that, in general, the difference in cosine similarity between synonyms and other similar words (from the top 10) is greater compared to, say, co-hyponyms. We also found this bigger difference for inflections and contrastives. This observation could be used to increase precision, as we can possibly filter out some co-hyponyms, related words, and unknowns.

To test this hypothesis, we developed a different measure to calculate similarity. We calculate similarity relative to the top n most similar words. We calculate *relative cosine similarity* between word w_i and w_j as in Equation 1.

$$rcs_n(w_i, w_j) = \frac{cosine\_similarity(w_i, w_j)}{\sum_{w_c \in TOP_n} cosine\_similarity(w_i, w_c)} \tag{1}$$

This will give words that have a high cosine similarity compared to other words in the top 10 most-similar words a high score. If all words in the top 10 most-similar words have almost an equal cosine similarity, they will get a lower score. When we do the same experiment again, changing the similarity threshold and plotting precision, recall and f-measure, using relative cosine similarity instead, we can see that precision goes up when we increase the rcs-threshold. This is shown in Figure 5. In Figure 6, it can also be noticed that when we look at the relative cosine similarity for the three most-similar words of words where the most similar word is synonym (solid), or simply a related word (dashed), part of the synonyms is now separable from the related words by a horizontal line, i.e. an rcs-threshold. This confirms our earlier hypothesis that synonyms have a bigger difference in cosine similarity with respect to other similar words.

We used WordNet synonyms here to calculate precision, recall and f-measure, and find the optimal rcs_{10}-threshold. However, what can be noticed is that the tilting point for the precision to go up lies at an rcs_{10}-threshold of 0.10. This is not a coincidence, as 0.10 is also the mean of the relative cosine similarities for 10 words. If a word has an rcs_{10} higher than 0.10, it is more similar than an arbitrary similar word. If synonyms are more similar compared to other similar word relations, we can find this tilting point at $\frac{1}{n}$, where n is the number of most-similar words we consider for calculating rcs_n.

Thus relative cosine similarity gives us the flexibility to increase precision, at the cost of recall, if needed. We can also identify the tilting point for precision to increase. For English and German this tilting point appears to lie at approximately the same threshold value. This will be shown in the next section, particularly in Figure 7.

3.6. Overlap of Similar Words in Different Vector Spaces

In this section, we explore whether we could use a combination of different vector spaces, trained using different training parameters to improve the synonym extraction. For this we analyze the most-cosine-similar words of the vocabulary S_{train} in different vector spaces. We considered pairs of vector

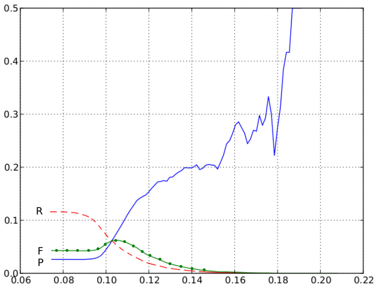


Figure 5. Precision, recall, and f-measure on S_{train} plotted against the relative cosine similarity threshold.

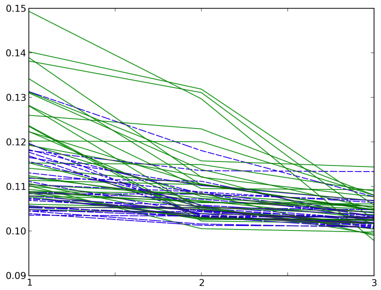


Figure 6. Relative cosine similarity against n-most similar position, for the 3-most-similar words, where the most-similar word is a synonym or a related word (dashed).

spaces with different training parameters. Then, we calculated the probability that an arbitrary word is most-cosine-similar in both vector spaces ($P(both)$). We also calculated the probability that a synonym is most-cosine-similar in both vector spaces ($P(both|synonym)$). We altered the dimension, window size and model (CBoW vs. SG). We mostly considered CBoW vectors, as they gave highest precision in previous experiments. The results of this experiment are shown in Table 7. What can be seen in this table is that for all changes in

Constant	Varies	$P(b)$	$P(b\|syn)$	$P(b\|syn) - P(b)$
CBoW win. 2	dim. 300 & 600	0.38	0.67	0.29
CBoW dim. 600	win. 2 & 4	0.31	0.60	0.30
CBoW dim. 600	win. 4 & 8	0.32	0.60	0.28
CBoW dim. 600	win. 2 & 8	0.24	0.52	0.28
dim. 300 win. 2	CBoW & SG	0.19	0.48	0.29

Table 7. Overlap between differently trained pairs of vector spaces, for arbitrary words, and synonyms. $P(b)$ is the probability of a word pair being most-similar in both vector spaces, $P(b|syn)$ is conditioned on the word being synonym.

parameters $P(\mathrm{both}|\mathrm{synonym})$ is considerably higher than $P(\mathrm{both})$. This indicates that it can be a good cue for synonymy if a word is most-cosine-similar in differently trained vector spaces. We can also see that the general overlap seems highest when only changing the number of dimensions, and lowest when changing the model, and fairly constant when doubling the window size. For all conditions, $P(\mathrm{both}|\mathrm{synonym}) - P(\mathrm{both})$ is fairly constant. This indicates that the cue for synonymy is almost equal for all pairs.

Because the numbers seem quite constant, it may be due to the inflections that overlap between both vector spaces. For this reason we repeated the experiment, but only considering word-pairs that have a Levenshtein distance greater than 3, to exclude the majority of the inflections. The results are shown in Table 8. Here we can see that the conclusion from Table 7 also holds for non-inflections. So, it is not just the inflections that overlap.

Constant	Varies	$P(b)$	$P(b|syn)$	$P(b|syn) - P(b)$
CBoW win. 2	dim. 300 & 600	0.31	0.61	0.30
CBoW dim. 600	win. 2 & 4	0.23	0.55	0.32
CBoW dim. 600	win. 4 & 8	0.24	0.56	0.32
CBoW dim. 600	win. 2 & 8	0.17	0.48	0.31
dim. 300 win. 2	CBoW & SG	0.12	0.42	0.30

Table 8. Overlap between differently trained pairs of vector spaces, for arbitrary words, and synonyms, when only considering word-pairs with a ***Levenshtein distance larger than 3.*** $P(b)$ *is the probability of a word pair being most-similar in both vector spaces,* $P(b|syn)$ *is conditioned on the word being synonym.*

To use this observations in our earlier synonym extraction method we calculate rcs_{10}^{m} in each vector space m for the 10 most-cosine-similar words on S_{train} in each space, and simply sum the rcs_{10} of the different models. The *summed relative cosine similarity* between word w_i and w_j is calculated in Equation 2, where $TOP_{10}^{m}(w_i)$ is the set containing the 10 closest cosine-similar words of w_i in vector space m.

$$rcs_{10}^{M} = \sum_{m}^{M} \begin{cases} rcs_{10}^{m}(w_i, w_j) & \text{if } w_j \in TOP_{10}^{m}(w_i) \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

As in the previous section, we again plot precision, recall, and f-measure against the threshold, but now using the summed rcs_{10} of a CBoW model, and a SG model. We did this for both German and English. For English, the CBoW model has 600 dimensions, and was trained with a window size of 4. The SG model has 150 dimensions, and a window size set to 2. For German, the CBoW model has 600 dimensions as well, and but a window size of 8. The results are shown in Figure 7. If we compare it to the results from Figure 5, we can

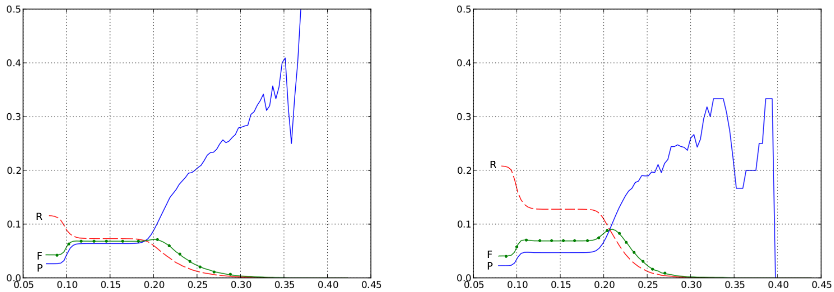


Figure 7. Precision, recall, and f-measure, on S_{train} for English (left) and German (right), using the summed rcs_{10} score for a CBoW and SG model.

see that for English, the general precision, recall, and f-measure lies higher using two vector spaces. Also, we can see that the tilting point now lies at around 0.2 instead of 0.1. It lies twice as high, as we sum rcs_{10} of two spaces. Also, our expectation that for different languages this tilting point lies at the same threshold seems correct for German. The bump in both graphs around a threshold of 0.1 shows up because some words only occur in the top-10 most similar words in one of the two vector spaces.

When we choose the threshold that gives optimal f-measure on the S_{train}, and use it to extract synonyms for S_{test}, we find for English a WordNet precision of 0.12, a recall of 0.05, and an f-measure of 0.07. Compared to our baseline of only taking the most similar word, precision is 1% absolute higher, recall is 2% higher, and f-measure 1%. For German, we find a precision of 0.12, recall of 0.07, and f-measure of 0.09. Compared to the baseline, precision went up with 4% absolute, recall with 2%, and f-measure with 3%. From this, we conclude that combining differently trained models helps to extract synonyms, both in precision, and recall. Also, combining the scores from the different vector spaces does not prevent us from finding the tilting point where precision rises.

4. Adding Parts-of-Speech

We now look at using a part-of-speech (POS) tagger to improve the synonym extraction in various ways.

4.1. Homography

The initial motivation to resort to POS-tagging is *homography*, i.e. one word (here, string of non-space characters) having several word-senses. In Figure 8, an example of homography of the words <phone> and <call> is given. The word senses and their respective parts of speech are shown in the leaves of the tree. The dotted link represents the synonym relation between the word-senses of <phone> and <call> for the action of making a telephone call.

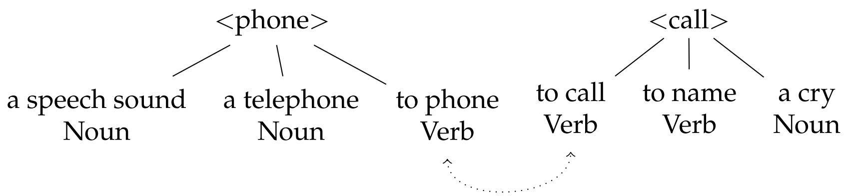


Figure 8. Schematic representation of the synonym relation between the corresponding word senses of the words <phone>, and <call>.

Homography can be a problem for finding synonyms when using one vector for each word, as the vector for <phone> is trained on all the different word-senses that occur in the corpus. In the case of <phone>, it is probably used more frequently as the noun telephone, or as a verb for the action of calling, compared to the noun meaning of a speech sound, in our news corpus. This can make it difficult to find synonyms with regard to this less frequent meaning.

To train vector representations for each word sense, ideally we would disambiguate each word in the corpus first, and then train the vectors on these disambiguated meanings. To our knowledge, there is not yet the possibility to do completely unsupervised word sense disambiguation. As can be seen in the example in Figure 8, some of the word senses can be separated by their parts of speech. We experimented with this, since POS tagging is available for many languages, and there are also options for word clustering/unsupervised POS-tagging (Christodoulopoulos et al., 2010).

4.2. Simple Part-of-Speech Tagging

In order to separate some word senses we preprocessed both the English and German corpora from the previous chapter with the Stanford POS tagger (Toutanova et al., 2003), using the fastest tag-models. Afterwards, we conflated the POS tags to five categories: (1) nouns, (2) verbs, (3) adjectives, (4) adverbs, and (5) the rest (no tag). An example of what the text looks like after tagging and simplification is given in Sentence 1.

1. Every day_N , I walk_V my daily_Adj walk_N .

In the example we can see that walk_V is distinct from walk_N, which will give us two different vectors. We chose these four tags as they correspond to the POS tags provided in WordNet and GermaNet. In this way, we can have a straightforward way to evaluate on the vocabulary (e.g. S_{train}). For each word, we now evaluate with regard to the synonyms that have the same POS in WordNet or GermaNet.

Another advantage of having these simple POS tags is that we can filter bad synonyms from the 10-most cosine similar words. Synonyms are very similar also on a grammatical level, as they are interchangeable in many contexts, so they should be of the same part-of-speech.

Because the vocabulary has changed, and the average frequency of words is now lower—as some words are split—we again analyze what word vector training parameters work best. We train CBoW and Skip-gram vectors on the tagged corpus, varying the dimensions over {150, 300, 600}, and the contextual window size over {2, 4, 16, 32}. We calculate precision for the most-similar and second-most-similar word for all words in S_{train}. The results are shown in Tables 9, 10, 11, and 12.

CBoW (Tagged)															
dim.	150					300					600				
win.	2	4	8	16	32	2	4	8	16	32	2	4	8	16	32
P-1	0.079	0.080	0.073	0.067	0.060	0.084	**0.085***	0.080	0.074	0.066	0.084	0.084	0.081	0.073	0.069
P-2	0.058	0.056	0.053	0.049	0.045	0.061	0.061	0.059	0.055	0.050	0.061	0.062	0.059	0.055	0.053

*Table 9. Precision for different window sizes and number of dimensions, using the **CBoW** model, for POS-tagged **English**.*

Skip-gram (Tagged)															
dim.	150					300					600				
win.	2	4	8	16	32	2	4	8	16	32	2	4	8	16	32
P-1	0.068	0.065	0.057	0.049	0.045	0.069	0.066	0.057	0.052	0.046	0.067	0.062	0.052	0.046	0.041
P-2	0.050	0.047	0.041	0.038	0.036	0.050	0.047	0.042	0.038	0.035	0.050	0.045	0.038	0.034	0.031

*Table 10. Precision for different window sizes and number of dimensions, using the **Skip-gram** model, for POS-tagged **English**.*

CBoW (Tagged)															
dim.	150					300					600				
win.	2	4	8	16	32	2	4	8	16	32	2	4	8	16	32
P-1	0.086	0.092	0.094	0.092	0.090	0.092	0.100	0.100	0.099	0.094	0.090	0.102	**0.103***	0.101	0.101
P-2	0.060	0.065	0.066	0.065	0.063	0.065	0.069	0.072	0.070	0.069	0.064	0.070	0.072	0.071	0.071

*Table 11. Precision for different window sizes and number of dimensions, using the **CBoW** model, for POS-tagged **German**.*

Skip-gram (Tagged)															
dim.	150					300					600				
win.	2	4	8	16	32	2	4	8	16	32	2	4	8	16	32
P-1	0.084	0.085	0.086	0.082	0.080	0.085	0.085	0.083	0.077	0.077	0.082	0.079	0.072	0.066	0.065
P-2	0.059	0.061	0.061	0.059	0.058	0.061	0.063	0.059	0.057	0.056	0.058	0.059	0.053	0.049	0.047

*Table 12. Precision for different window sizes and number of dimensions, using the **Skip-gram** model, for POS-tagged **German**.*

If we look at Table 9 we can see that the highest precision is obtained using a CBoW model with a window size of 4, and 600 dimensions. If we compare this to the best results on the non-tagged corpus, from Table 2 in Section 3, the optimal window size has stayed the same. Also CBoW vectors still perform better than Skip-gram vectors, and small windows work best for Skip-gram vectors. However, the best performing number of dimensions went from 600 to 300 when adding the POS-tag for English. A possible explanation can be that since the part-of-speech tags separate some of the word contexts, based on grammatical properties, the same information can be encoded with less dimensions.

For German, precision went up when adding the POS-tags. This can be seen if we compare the precision from Tables 4 and 5 with Tables 11 and 12. The best vectors are still CBoW vectors with 600 dimensions and a contextual window of 8. When we tried to find the reason why German has such a increase in precision compared to English, we found that it lies partially at the level of POS-tag simplification. As in the German part-of-speech tagset, the *Stuttgart-Tübingen tagset* (STTS), names are not considered as nouns. For this reason we did not conflate them to a noun tag, and they were excluded during evaluation. This was not the case for English. Names are one of the frequent error categories we found in Section 3.

This highlights another use of the POS tagger, which is that we can simply exclude categories for which we don't want to find synonyms, and maybe even filter bad synonym candidates from the 10-most-similar words. An example would be the frequent error category of plurals, but also other types of inflections, which can be filtered, as they are given a different POS tag (before tag conflation). These insights will be used in the final system, presented in Section 5.

To compare using the simplified POS tags with the previous approaches we also calculated precision, recall and f-measure on S_{test}. Compared to the baseline of looking only at the most-similar word, we found that recall in English increased from 3% to 4%, precision did not change (11%), and f-measure from 5% to 6%. Notably, German precision increased with 8% to 12%, recall from 5% to 7%, and f-measure from 6% to 9%.

From these experiments we conclude that POS tags can help to improve synonym extraction in three ways. Firstly, they can separate some of the word senses, however this effect is minor. Secondly, they can filter words that are not grammatically similar enough, such as plurals. And thirdly, they can exclude synonyms in categories for which there no, or very few, synonyms, such as names.

5. Final System and Evaluation

In this section we describe and evaluate the final systems for English and German that we constructed from the findings from the previous sections.

5.1. The System

For the final systems we used larger corpora than those used in the previous experiments. We used 500 million tokens from the same corpora as before, the English and German NewsCrawl 2014 corpora from the Workshop on Machine Translation in 2015. We POS tagged the corpora using the same parser and models as in Section 4. However, we do not simplify the POS tags, but instead use the fine-grained tags for nouns, verbs, adjectives or adverbs. We exclude the tags for names, as they have few to no synonyms.

It should be noted that in the German tagset there is only one tag for nouns, which covers both singular and plural nouns. This might result in more errors. For machine translation evaluation we do not expect this to have a large negative impact, as plurals would also have been matched by Meteor in the stemming module. However, it might result in a worse human evaluation.

For English we train CBoW vectors with 300 dimensions and a contextual window of 4. We also train Skip-gram vectors with 300 dimensions and a contextual window of 2. For German we train vectors with the same specifications, except for the German CBoW model we use a contextual window of size 8, and for Skip-gram a window of size 4. We chose these parameter settings as a compromise between the optimal parameters from our experiment in Chapter 4, and our expectations with respect to introducing fine-grained POS tags, which is that the optimal number of dimensions might decrease slightly.

We only consider words that occur at least 20 times in the corpus. The reasons for using a higher frequency threshold are (1) to obtain better quality word vectors, as we aim for high precision, and (2) to maintain a vocabulary size similar to the previous experiments, as we increased corpus size. The resulting tagged English vocabulary contains 115,632 word types, and the German vocabulary 311,664.

We then calculate the summed relative cosine similarity of both the CBoW and the Skip-gram vectors for the full vocabulary with regard to the top-10 most cosine-similar words. We select word pairs with a summed rcs_{10} similarity higher than 0.22. We choose 0.22 as it lies slightly above the expected tilting point of 0.2. For English, we obtain 16,068 word pairs. For German we obtain 96,998 word pairs. It should be noted that the word pairs are also tagged, which can be useful depending on the application.

5.2. Manual Evaluation

To evaluate the precision of the obtained synonyms, we took a random sample of 200 word pairs for both languages. The word pairs were then annotated for synonymy. The annotation categories are synonyms, non-synonyms, or unknown. In the description the unknown category is indicated for when an annotator does not know any of the two words. The annotators could also indicate hesitation, but still had to give a preference for any of the three categories.

For English, annotation is done by two annotators. One annotator is a native English speaker and one a fluent non-native speaker. For German, annotation is also done by two annotators, one native German speaker, and one an intermediate non-native speaker. Annotators could access the internet to look up synonymy, or word meanings. We discriminate several situations:

SS: Both annotators annotate synonymy
NN: Both annotators annotate non-synonymy
SU: One annotator annotates synonymy, and the other unknown
NU: One annotator annotates non-synonymy, and the other unknown
SN: One annotator annotates synonymy, and the other non-synonymy
UU: Both annotators annotate unknown

We assume that if both annotators do not know the words, there is no synonymy. We can calculate a *lower bound of precision* (P^{-}_{syn}), and an *upper bound of precision* (P^{+}_{syn}). For the lower bound, we only consider word pairs of category SS as synonyms, and the rest as non-synonyms. For the upper bound, we consider word pairs of category SS and SU as synonyms, and the rest as non-synonyms.

We also calculate a lower and upper bound for non-synonymy ($P^{-}_{\neg syn}$, and $P^{+}_{\neg syn}$), and the percentage of disagreement on the categories of synonym and non-synonym ($P_{disagree}$). This way we can get a better idea of how many clear errors there are, and how many errors are unclear.

The results for both English and German are shown in Table 13. What can be noticed is that for German, the precision is quite a bit lower than for English. However, the number of found word pairs is much higher. One reason can be that the threshold should be higher in order to get comparable precision. A second reason can be that for English the error categories, such as plurals, are separated by a POS tag, resulting in higher precision. In the German tagset these are not separated. We found that 10% of the German word pairs in this

Manual Evaluation	P^{-}_{syn}	P^{+}_{syn}	$P^{-}_{\neg syn}$	$P^{+}_{\neg syn}$	$P_{disagree}$	P_{UU}
English	0.55	0.59	0.15	0.21	0.16	0.05
German	0.30	0.35	0.42	0.49	0.15	0.03

Table 13. Manual evaluation of the final systems.

set are plurals. For English, there were no such cases. For our application, these errors should not be a major problem, as plurals would otherwise have been matched by the stemming module of Meteor.

The percentage of unknown words seems fairly small, and about the same for both languages. Also the disagreement on synonymy seems about the same for both languages, around 15%. The cause for disagreement could be the difference in the language level of the speakers. Another reason could be the subjectivity of the notion of synonymy.

5.3. Application in Machine Translation Evaluation

To see if the quality of the extracted synonyms is sufficient for the synonyms to be beneficial in an application we also used them in machine translation evaluation. We use them in the synonym module of the Meteor 1.5 evaluation metric.

We use the synonyms extracted by the system described in Section 5.1. So for German, the synonym resource will consist of the 96,998 word pairs, and for English we use 16,068 word pairs.

Meteor weighs the scores from each matching module. For English, we use the default weights (Denkowski and Lavie, 2014), as synonyms were already incorporated for English. For German, we use the default weights for all other modules, except we use the same weight for the synonym module as used for English (0.80).

To evaluate the metric, we test if the Meteor score correlates better with human judgments after adding our synonyms. We calculate the correlation using the data from the metrics task of the workshop on machine translation 2014[4] (WMT 2014) (Macháček and Bojar, 2014).

We use the news-test reference sentences from the language pair German-English, for English. This set consists of around 3000 segments, or sentences. For German, we use the reference sentences from the English-German language pair. This set consists of around 2700 segments, or sentences.

[4]http://www.statmt.org/wmt14/results.html

We calculate *segment-level Kendall's* τ *correlation* as calculated in the WMT 2014 for the following three Meteor conditions:

1. Using all four modules, with the default weights, and no synonym resource.
2. Using all four modules, using default weights, and with our synonyms.
3. Using all four modules, using default weights, using WordNet synonyms (only for English).

Kendall's τ is expected to predict the result of the pairwise comparison of two translation systems. In WMT-2014 this is calculated using human judgments on a ranking task of 5 systems per comparison. τ is calculated as in Equation 3, where *Concordant* is the set of human comparisons for which the Meteor score suggests the same order, and *Discordant* is the set of all human comparisons for which a given metric disagrees. When the Meteor score gives the same rankings as the human judgments, correlation will be high, and vice versa.

$$\tau = \frac{|Concordant| - |Discordant|}{|Concordant| + |Discordant|} \tag{3}$$

We calculated the Meteor scores for hypotheses from the 13 translation systems for the language pair German-English, and the 18 translation systems for English-German.

We also calculated the *system level correlation*, which indicates to what degree the evaluation metric orders the translation systems in the same order as the human judgments do, based on the total system score that the evaluation metric gives to each system. This is calculated as the *Pearson correlation*, as described by Macháček and Bojar (2014), and in Equation 4, where H is the vector of human scores of all systems translating in the given direction, M is the vector of the corresponding scores as predicted by the given metric, here Meteor. $\bar{H}$ and $\bar{M}$ are their means respectively.

$$r = \frac{\sum_{i=1}^{n}(H_i - \bar{H})(M_i - \bar{M})}{\sqrt{\sum_{i=1}^{n}(H_i - \bar{H})^2}\sqrt{\sum_{i=1}^{n}(M_i - \bar{M})^2}} \tag{4}$$

Both the segment-based correlations and the system-level correlations are shown in Table 14 for the same conditions as mentioned before. It can be seen that for both English and German using the extracted synonyms has a positive effect on both the segment correlation and the system correlation. It can also be noticed that using WordNet gives the highest correlation for English.

From this we conclude that currently our method, using only raw text and a POS tagger, does not outperform a large manually constructed synonym

German-**English**	τ	r
Condition 1	0.323	0.915
Condition 2	0.326	0.917
Condition 3	0.334	0.927

English-**German**	τ	r
Condition 1	0.238	0.263
Condition 2	0.243	0.277
Condition 3	-	-

Table 14. System level correlations (r), and segment level correlations (τ) for the Meteor 1.5 score without synonyms (condition 1), when adding the extracted synonyms (condition 2), and when using WordNet synonyms (condition 3).

database such as WordNet, but can be useful to extract synonyms when no such resource is available for the target language in Meteor[5].

What should be noted is that the extracted synonyms are not yet fully exploited, as Meteor ignores the POS tags that were given to the synonyms. If two words are synonymous with respect to their part of speech, but not synonymous if they are of different parts of speech, Meteor will align them in both situations. In the case when the words are of different POS, they will be falsely aligned by Meteor.

The improvement of the metric is greater for German than for English. This might seem odd at first, since the German synonyms had a lower precision in manual evaluation compared to the English synonyms. But still, they perform better in machine translation evaluation. This can be explained by what was already mentioned earlier, that a significant part of the German synonym errors are inflections, due to the difference in POS tagset. Also, the synonyms extracted for German are less ambiguous with respect to their part of speech. The German language frequently employs compounding (e.g. *Schwierigkeitsgrade*, ‘degree of difficulty’), and grammatical case markers. This might result in less ambiguous words. The negative effect of Meteor not using parts of speech with synonyms could be smaller for German for this reason. Furthermore, the difference could also be explained by the difference in the number of synonyms (~16K for English, and ~97K for German).

6. Conclusions & Future Work

In this article we explored different methods to extract synonyms from text. The initial motivation was to use the extracted synonyms to improve machine translation evaluation. We tried to extract the synonyms using as little su-

[5] Our German results are an indirect example of this: even though a WordNet resource (GermaNet) exists, it is not available to Meteor due to licencing reasons.

pervision as possible, with the goal that the same method can be applied to multiple languages. We experimented with English and German.

Word vectors trained using the continuous bag-of-words model (CBoW), and the skip-gram model (SG) proposed by Mikolov et al. (2013a) were used in the experiments. We evaluated different hyperparameters for training these vectors for synonym extraction. In our experiments CBoW vectors gave higher precision and recall than SG vectors. The number of dimensions did not seem to play a very large role. For our experiments, dimensions of 300 and 600 seemed to give best results. The optimal contextual windows size was around 4 for English and 8 for German. We hypothesized that the difference in window size can be because of the difference in the distributions of word categories of the synonyms in WordNet and GermaNet.

For English, we manually looked at frequent error categories when using these vectors for this task. The largest well-defined error categories we found are inflections, co-hyponyms, and names.

We found that the cosine similarity on its own is a bad indicator to determine if two words are synonymous. We proposed *relative cosine similarity*, which calculates similarity relative to other cosine-similar words in the corpus. This is a better indicator, and can help improve precision. Also, the optimal thresholds for finding synonyms for English and German using this measure are almost the same. This gives hope for easy extension of this method to other languages, for which there is no synonym data. It would be very interesting to see to which other languages this method can generalize.

We also experimented with combining similarity scores from differently trained vectors, which seems to slightly increase both precision and recall. Furthermore, we explored the advantages of using a POS tagger as a way of introducing some light supervision. POS tags can help performance in different ways. Firstly, it can disambiguate some of the meanings of homographs. Secondly, it can help filter bad synonym candidates. And thirdly, it can prevent extraction of synonyms for word categories that have no, or very few synonyms, such as names. For future research, it would be interesting to examine the effect of using an unsupervised POS tagger (Christodoulopoulos et al., 2010).

We could also investigate the use of topical word embeddings (Liu et al., 2015a,b), or global context vectors (Huang et al., 2012). These techniques make different vectors for each word using topical information to disambiguate some of the different word senses.

We evaluated our final approach for both English and German. We did a manual evaluation with two annotators per language. We also applied the extracted synonyms in machine translation evaluation. From the manual evaluation, the English synonyms had higher precision than the German ones. A likely reason for this is that the English POS tagset better separates the frequent error categories mentioned in Section 3.

When we evaluated the quality of the extracted synonyms in the task of machine translation evaluation (with the Meteor metric) for both English and German, the extracted synonyms increased the correlation of the metric with human judgments, resulting in an improved evaluation metric. While our method currently does not outperform a manually constructed synonym database such as WordNet, it can be useful to extract synonyms when no such resource is available for the target language, or domain. As the method uses tokenized raw text and optionally a POS tagger, it is applicable to a wide range of languages.

In the current research, we used a fixed frequency threshold, excluding infrequent words (a large part of the vocabulary). Setting a threshold also influences the word embedding training. For future research, it would be interesting to see the impact of the frequency threshold on our method.

Moreover, currently Meteor does not fully exploit the extracted synonyms, as it ignores their POS, which can cause false alignments. For future research on improving Meteor, it could be interesting to incorporate POS tags to prevent inappropriate generalization of synonyms.

Acknowledgements

The authors would like to thank the anonymous reviewers for their valuable comments and suggestions to improve the quality of the paper.

Bibliography

Agirre, Eneko, Enrique Alfonseca, Keith Hall, Jana Kravalova, Marius Paşca, and Aitor Soroa. A Study on Similarity and Relatedness Using Distributional and WordNet-based Approaches. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 19–27, Boulder, CO, USA, June 2009. Association for Computational Linguistics.

Banerjee, Satanjeev and Alon Lavie. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, volume 29, pages 65–72, 2005.

Christodoulopoulos, Christos, Sharon Goldwater, and Mark Steedman. Two Decades of Unsupervised POS induction: How far have we come? In *Proceedings of the*

2010 Conference on Empirical Methods in Natural Language Processing, pages 575–584. Association for Computational Linguistics, 2010.

Comelles, Elisabet and Jordi Atserias. VERTa Participation in the WMT14 Metrics Task. In *Proceedings of the 9th Workshop on Statistical Machine Translation (WMT)*, pages 368–375, Baltimore, Maryland, USA, June 2014. Association for Computational Linguistics.

Denkowski, Michael and Alon Lavie. Meteor Universal: Language Specific Translation Evaluation for Any Target Language. In *Proceedings of the EACL 2014 Workshop on Statistical Machine Translation*, 2014.

Doddington, George. Automatic Evaluation of Machine Translation Quality Using N-gram Co-Occurrence Statistics. In *Proceedings of the 2nd International Conference on Human Language Technologies (HLT)*, pages 138–145, 2002.

Faruqui, Manaal, Jesse Dodge, Sujay Kumar Jauhar, Chris Dyer, Eduard Hovy, and Noah A. Smith. Retrofitting Word Vectors to Semantic Lexicons. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1606–1615, Denver, CO, USA, 2015. Association for Computational Linguistics.

Fu, Ruiji, Jiang Guo, Bing Qin, Wanxiang Che, Haifeng Wang, and Ting Liu. Learning semantic hierarchies via word embeddings. In *Proceedings of the 52th Annual Meeting of the Association for Computational Linguistics: Long Papers*, volume 1, 2014.

Gonzà lez, Meritxell, Alberto Barrón-Cedeño, and Lluís Màrquez. IPA and STOUT: Leveraging Linguistic and Source-based Features for Machine Translation Evaluation. In *Proceedings of the 9th Workshop on Statistical Machine Translation (WMT)*, pages 394–401, Baltimore, Maryland, USA, June 2014. Association for Computational Linguistics.

Gupta, Dishan, Jaime Carbonell, Anatole Gershman, Steve Klein, and David Miller. Unsupervised Phrasal Near-Synonym Generation from Text Corpora. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015a.

Gupta, Rohit, Constantin Orăsan, and Josef van Genabith. ReVal: A Simple and Effective Machine Translation Evaluation Metric Based on Recurrent Neural Networks. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Lisbon, Portugal, 2015b.

Harris, Zellig S. Distributional Structure. *Word*, 1954.

Huang, Eric H., Richard Socher, Christopher D. Manning, and Andrew Y. Ng. Improving word representations via global context and multiple word prototypes. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 873–882. Association for Computational Linguistics, 2012.

Levy, Omer and Yoav Goldberg. Dependency-based word embeddings. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, volume 2, pages 302–308, 2014.

Lin, Dekang, Shaojun Zhao, Lijuan Qin, and Ming Zhou. Identifying synonyms among distributionally similar words. In *IJCAI*, volume 3, pages 1492–1493, 2003.

Liu, Pengfei, Xipeng Qiu, and Xuanjing Huang. Learning Context-Sensitive Word Embeddings with Neural Tensor Skip-Gram Model. *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence*, 2015a.

Liu, Yang, Zhiyuan Liu, Tat-Seng Chua, and Maosong Sun. Topical Word Embeddings. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015b.

Luong, Minh-Thang, Richard Socher, and Christopher D. Manning. Better word representations with recursive neural networks for morphology. *CoNLL-2013*, 104, 2013.

Macháček, Matouš and Ondřej Bojar. Results of the WMT14 metrics shared task. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 293–301, Baltimore, MD, USA, 2014.

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

Mikolov, Tomáš, Ilya Sutskever, Kai Chen, Greg S. Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013b.

Miller, George A. WordNet: a lexical database for English. *Communications of the ACM*, 38(11):39–41, 1995.

Ono, Masataka, Makoto Miwa, and Yutaka Sasaki. Word Embedding-based Antonym Detection using Thesauri and Distributional Information. In *Proc. of NAACL*, 2015.

Pak, Alexander Alexandrovich, Sergazy Sakenovich Narynov, Arman Serikuly Zharmagambetov, Sholpan Nazarovna Sagyndykova, Zhanat Elubaevna Kenzhebayeva, and Irbulat Turemuratovich. The Method of Synonyms Extraction from Unannotated Corpus. In *Digital Information, Networking, and Wireless Communications (DINWC), 2015 Third International Conference on*, pages 1–5. IEEE, 2015.

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

Pennington, Jeffrey, Richard Socher, and Christopher D. Manning. GloVe: Global vectors for word representation. *Proceedings of the Empiricial Methods in Natural Language Processing (EMNLP 2014)*, 12, 2014.

Plas, Lonneke van der and Gosse Bouma. Syntactic contexts for finding semantically related words. *LOT Occasional Series*, 4:173–186, 2005.

Saveski, Martin and Igor Trajkovski. Automatic construction of wordnets by using machine translation and language modeling. In *13th Multiconference Information Society, Ljubljana, Slovenia*, 2010.

Snover, Matthew, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. A study of translation edit rate with targeted human annotation. In *Proceedings of Association for Machine Translation in the Americas*, pages 223–231, 2006.

Stanojević, Miloš and Khalil Sima'an. BEER: BEtter Evaluation as Ranking. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 414–419, Baltimore, Maryland, USA, June 2014. Association for Computational Linguistics.

Tan, Liling, Rohit Gupta, and Josef van Genabith. USAAR-WLV: Hypernym Generation with Deep Neural Nets. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, 2015.

Toutanova, Kristina, Dan Klein, Christopher D Manning, and Yoram Singer. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, pages 173–180. Association for Computational Linguistics, 2003.

Van der Plas, Lonneke and Jörg Tiedemann. Finding synonyms using automatic word alignment and measures of distributional similarity. In *Proceedings of the COLING/ACL on Main conference poster sessions*, pages 866–873. Association for Computational Linguistics, 2006.

van der Plas, Lonneke, Joerg Tiedemann, and Jean—Luc Manguin. Automatic acquisition of synonyms for French using parallel corpora. *Distributed Agent-based Retrieval Tools*, page 99, 2010.

Vela, Mihaela and Ekaterina Lapshinova-Koltunski. Register-Based Machine Translation Evaluation with Text Classification Techniques. In *Proceedings of the 15th Machine Translation Summit*, pages 215–228, Miami, Florida, November 2015. Association for Machine Translations in the Americas.

Vela, Mihaela and Liling Tan. Predicting Machine Translation Adequacy with Document Embeddings. In *Proceedings of the 10th Workshop on Statistical Machine Translation (WMT)*, pages 402–410, Lisbon, Portugal, September 2015. ACL.

Vossen, Piek. *A multilingual database with lexical semantic networks*. Kluwer Academic Publishers, Dordrecht, 1998.

Yu, Hong, Vasileios Hatzivassiloglou, Carol Friedman, Andrey Rzhetsky, and W John Wilbur. Automatic extraction of gene and protein synonyms from MEDLINE and journal articles. In *Proceedings of the AMIA Symposium*, page 919. American Medical Informatics Association, 2002.

Address for correspondence:
Artuur Leeuwenberg
tuur.leeuwenberg@cs.kuleuven.be
Katholieke Universiteit Leuven, Department of Computer Science
Celestijnenlaan 200A, 3000 Leuven, Belgium

PBML

The Prague Bulletin of Mathematical Linguistics

NUMBER 105 APRIL 2016 143–193

Universal Annotation of Slavic Verb Forms

Daniel Zeman

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

Abstract

This article proposes application of a subset of the Universal Dependencies (UD) standard to the group of Slavic languages. The subset in question comprises morphosyntactic features of various verb forms. We systematically document the inventory of features observable with Slavic verbs, giving numerous examples from 10 languages. We demonstrate that terminology in literature may differ, yet the substance remains the same. Our goal is practical. We definitely do not intend to overturn the many decades of research in Slavic comparative linguistics. Instead, we want to put the properties of Slavic verbs in the context of UD, and to propose a unified (Slavic-wide) application of UD features and values to them. We believe that our proposal is a compromise that could be accepted by corpus linguists working on all Slavic languages.

1. Introduction and related work

Universal Dependencies (Nivre et al., 2016)[1] is a project that seeks to design cross-linguistically consistent treebank annotation for as many languages as possible. Besides dependency relations, UD also defines universally applicable tags for parts of speech *(universal POS tags)* and common morphosyntactic features *(universal features).* The features are taken from a previous project called Interset (Zeman, 2008).

Being suitable for a variety of unrelated languages means that the core concepts of UD must be sufficiently general; at the same time, their definitions must be descriptive enough to signal that two phenomena in two different languages are (or are not) the same thing, despite conflicts in traditional terminologies.

There is always the danger that researchers working on different languages will apply the UD concepts differently. As UD gains on popularity and new datasets are

[1] `http://universaldependencies.org/`

 Corresponding author: zeman@ufal.mff.cuni.cz
Cite as: Daniel Zeman. Universal Annotation of Slavic Verb Forms. The Prague Bulletin of Mathematical Linguistics No. 105, 2016, pp. 143–193. doi: 10.1515/pralin-2016-0007.

converted to its annotation scheme, enforcing consistency is an increasingly important issue. It seems natural to start with looking at closely related languages and first make sure that they annotate the same things same way; then widen the view to larger language groups and so on.

The first work on Slavic-specific issues in UD was Zeman (2015). The present article focuses on part-of-speech tags and features of individual words, not on inter-word dependency relations. Some verb forms are analytical (periphrastic), made of two or more individual words. We occasionally use the periphrastic constructions for illustrative purposes but bear in mind that tags and features must be assigned to individual words only. Also note that UD postulates the concept of syntactic word, something that is not necessarily identical to the space-delimited orthographic word. An orthographic word may be understood as a fusion of two or more syntactically autonomous units; the annotation treats each of them separately.

Some work has been published that pre-dates UD and is related to our current effort. Besides Interset (Zeman, 2015), the outcomes of the MULTEXT-East project are highly relevant (Erjavec, 2012). Quite a few Slavic languages have morpho-syntactic tagsets stemming from MULTEXT-East. These tagsets are similar to each other and they were indeed intended to encode the same phenomena identically across languages. Unfortunatelly they have not always reached this goal. Traditional views and legacy resources sometimes outweighed the desire for uniformity. UD faces the same danger and we should strive hard to avoid it.

In the following sections we discuss UD tags and features applicable to Slavic verbs (as well as some words on the border between verbs and other parts of speech). We give numerous examples and inflection tables together with the proposed annotation.[2] We list the native names of the verb forms in the beginning of each section.

We use ISO 639 language codes when refering to individual languages: **[be]** Belarusian, **[bg]** Bulgarian, **[cs]** Czech, **[cu]** Old Church Slavonic, **[dsb]** Lower Sorbian, **[hr]** Croatian, **[hsb]** Upper Sorbian, **[mk]** Macedonian, **[pl]** Polish, **[ru]** Russian, **[sk]** Slovak, **[sl]** Slovenian, **[sr]** Serbian, **[uk]** Ukrainian.

Six Slavic languages ([bg], [cs], [cu], [hr], [pl] and [sl]) already have datasets in the current release of UD (1.2) and other languages are expected to get covered in the near future. We briefly summarize the approaches taken in the current data in Section 18.

2. Universal Features

The following universal features are discussed in the article. See the on-line documentation of UD (`http://universaldependencies.org/`) for their detailed description with examples. Here we provide just a list for quick reference:

[2]The tables were compiled using on-line resources such as Wictionary, verb conjugators and language courses, as well as printed grammars and dictionaries. We do not cite these sources individually due to space considerations.

- `Aspect`: `Imp` (imperfective), `Perf` (perfective)
- `VerbForm`: `Fin` (finite verb), `Inf` (infinitive), `Sup` (supine), `Part` (participle), `Trans` (transgressive)
- `Mood`: `Ind` (indicative), `Imp` (imperative), `Cnd` (conditional)
- `Tense`: `Past` (past), `Imp` (imperfect), `Pres` (present), `Fut` (future)
- `Voice`: `Act` (active), `Pass` (passive)
- `Number`: `Sing` (singular), `Dual` (dual), `Plur` (plural)
- `Person`: `1`, `2`, `3`
- `Gender`: `Masc` (masculine), `Fem` (feminine), `Neut` (neuter)
- `Animacy`: `Anim` (animate/human), `Nhum` (animate nonhuman), `Inan` (inanimate)
- `Case`: `Nom` (nominative), `Gen` (genitive), `Dat` (dative), `Acc` (accusative), `Voc` (vocative), `Loc` (locative), `Ins` (instrumental)
- `Definite`: `Ind` (indefinite), `Def` (definite)
- `Negative`: `Pos` (affirmative), `Neg` (negative)

3. Universal Part of Speech Tag and Lemma

We discuss various finite and non-finite forms of verbs in Slavic languages. We include some forms on the border of verbs and other parts of speech because we want to define the borderline between parts of speech uniformly for all Slavic languages.

We propose a simple (but approximate!) rule of thumb: if it inflects for `Case`, it is not a `VERB`. It is either an `ADJ`, or a `NOUN`. We treat such forms as adjectives or nouns derived from verbs. Nevertheless, they may have some features such as `VerbForm` and `Tense` that are normally used with verbs and that do not occur with other adjectives and nouns.

Verbal nouns have the neuter gender and they are rarely seen in plural.

Participles may, depending on language, have short and long forms. The long forms almost always inflect for `Case` and can be used attributively (as modifiers of nouns). We propose to classify them as adjectives. The short forms of some participle types receive the `VERB` tag: it signals that their inflection is limited[3] and their usage is prevailingly predicative. In south Slavic languages even some short participles inflect for `Case`[4] and get the `ADJ` tag; the short vs. long forms differ in the feature of `Definite`(ness) there.

Only a few Slavic verbs may function as auxiliaries and be tagged `AUX`. All of them may also be tagged `VERB` in other contexts. The main auxiliary verb is *to be (být, bývat, byť, być, бути, быть, biti…)* It may be used to form the future tense, past tense, conditional and passive. Serbo-Croatian languages use a different auxiliary verb, *htjeti*

[3]A rare example of short form inflection in Czech is the feminine accusative, e.g. *udělánu*.

[4]Actually only a few forms—masculine singular nominative and masculine inanimate singular accusative—distinguish “long” vs. “short” forms in [sl] and [hr]. In the other cases there is just one form and it does not make much sense to classify it as either long or short.

"will", to form the future tense. We do not see any benefit in granting the auxiliary status to verbs that are not needed in periphrastic verb forms; in particular, modal verbs are tagged `VERB`, although UD for Germanic languages treats them as auxiliaries. In accord with the UD guidelines, the verb *to be* is tagged `VERB` if it functions as copula.

All words tagged `VERB` or `AUX` must have a non-empty value of the feature `VerbForm`.

The POS tag also determines what word form will be used as the lemma. For `VERB` and `AUX`, the lemma is the infinitive (Section 5),[5] except for [bg] and [mk]: these languages do not have infinitives, and present indicative forms are used as lemmas there. However, if the word is tagged `ADJ`, the masculine singular nominative form of the adjective serves as the lemma. The annotation does not show the infinitive of the base verb (except for an optional reference in the `MISC` column). Similarly, the lemma of a verbal `NOUN` is its singular nominative form.

4. Aspect

Slavic languages distinguish two aspects: imperfective (`Aspect=Imp`) and perfective (`Aspect=Perf`). The feature is considered lexical, that is, all forms of one lemma (usually) belong to the same aspect. A few verbs (many of them loanwords from non-Slavic languages) work with both aspects. We omit the `Aspect` feature at these verbs. Most Slavic verbs are part of inflected aspect pairs where one verb is imperfective and the other is perfective. They have different lemmas and the morphological processes that create one from the other are considered derivational. Examples (`Imp – Perf`): [cs] *dělat – udělat* "to do", *sedět – sednout* "to sit", *kupovat – koupit* "to buy", *brát – vzít* "to take". Although the meaning of the two verbs is similar, in perfective verbs the action is completed and in imperfective verbs it is ongoing.

The equivalents of the verb *to be* are imperfective.

5. Infinitive and Supine

[cs] *infinitiv, neurčitek;* [sk] *infinitív, neurčitok;* [hsb] *infinitiw;* [pl] *bezokolicznik;* [uk] *інфінітив;* [ru] *инфинитив;* [sl] *nedoločnik* (`Inf`), *namenilnik* (`Sup`); [hr] *infinitiv.* Tables 1 and 2.

Most Slavic languages have a distinct infinitive form, which is used as argument of modal and other verbs (control, purpose), and sometimes in construction of the periphrastic future tense. The infinitive is also used as the citation form of verbs. It does not exist in Macedonian and Bulgarian.

Czech has two forms of infinitive, e.g. *dělat* and *dělati* "to do". The longer form with the final *-i* is considered archaic, otherwise they are grammatically equivalent.

[5] We do not prescribe whether inherently reflexive verbs such as [cs] *smát se* "to laugh" should or should not have the reflexive pronoun incorporated in their lemma.

en	*to be*	*can*	*to go*	*to do*	*to accept*
cs	*být, býti*	*moct, moci*	*jít, jíti*	*dělat, dělati*	*akceptovat, akceptovati*
sk	*byť*	*môcť*	*ísť*	*robiť*	*akceptovať*
hsb	*być*	*móc*	*hić*	*dźěłać*	*akceptować*
pl	*być*	*móc*	*iść*	*robić*	*akceptować*
uk	*бути* *buty*	*могти* *mohty*	*йти* *jty*	*робити* *robyty*	*акцептувати* *akceptuvaty*
ru	*быть* *byt'*	*мочь* *moč'*	*идти* *idti*	*делать* *delat'*	*акцептовать* *akceptovat'*
sl	*biti*	*moči*	*iti*	*delati*	*akceptirati*
hr	*biti*	*moći*	*ići*	*delati, delat*	*akceptirati, akceptirat*
cu	быти *byti*	мощи *mošti*	ити *iti*	дѣлати *dělati*	

Table 1. `VerbForm=Inf`

en	*to be*	*can*	*to go*	*to do*	*to accept*
sl	*bit*		*it*	*delat*	*akceptirat*
cu	бытъ *bytъ*		итъ *itъ*	дѣлатъ *dělatъ*	

Table 2. `VerbForm=Sup`

In contrast, Slovenian uses only the longer form *(delati)* as infinitive, while the shorter form is called supine and is used after motion verbs (meaning “to go somewhere to do something”).[6] In Croatian both are considered infinitive but the short form is only used in future tense if the infinitive precedes the auxiliary verb: ***Učit*** *ću hrvatski.* “I will learn Croatian.” but *Hoću* ***učiti*** *hrvatski.*

Infinitive and supine verbs lack most other verbal features, they only have non-empty values of `Aspect`, `VerbForm` and in some languages also of `Negative`.

6. Present and Future Indicative

[cs] *přítomný čas (prézens), budoucí čas (futurum);* [sk] *prítomný čas, budúci čas;* [hsb] *prezens, futur;* [pl] *czas teraźniejszy, czas przyszły;* [uk] *теперішній час, майбутній час;*

[6] The supine is an old form, attested in Old Church Slavonic. Besides Slovenian, it has also survived in Lower Sorbian.

Number	Sing			Dual			Plur		
Person	1	2	3	1	2	3	1	2	3
cs	*jsem*	*jsi*	*je*				*jsme*	*jste*	*jsou*
sk	*som*	*si*	*je*				*sme*	*ste*	*sú*
hsb	*sym*	*sy*	*je*	*smój*	*staj*	*staj*	*smy*	*sće*	*su*
pl	*jestem*	*jesteś*	*jest*				*jesteśmy*	*jesteście*	*są*
uk	*є* *je*	*єси, є* *jesy, je*	*є* *je*				*є* *je*	*є* *je*	*є* *je*
ru			*есть* *est′*						*суть* *sut′*
sl	*sem*	*si*	*je*	*sva*	*sta*	*sta*	*smo*	*ste*	*so*
hr	*jesam* *sam*	*jesi* *si*	*jest* *je*				*jesmo* *smo*	*jeste* *ste*	*jesu* *su*
bg	*съм* *săm*	*си* *si*	*е* *e*				*сме* *sme*	*сте* *ste*	*са* *sa*
cu	ЕСМЪ *jesmь*	ЕСИ *jesi*	ЕСТЪ *jestъ*	ЕСВѢ *jesvě*	ЕСТА *jesta*	ЕСТЕ *jeste*	ЕСМЪ *jesmъ*	ЕСТЕ *jeste*	СѪТЪ *sǫtъ*

Table 3. To be, `VerbForm=Fin | Mood=Ind | Tense=Pres`. Note that in Ukrainian and Russian the original non-3rd person forms of this verb have become archaic.

[ru] *настоящее время, будущее время;* [sl] *sedanjik, prihodnjik;* [hr] *sadašnje vrijeme, buduće vrijeme;* [bg] *сегашно време, бъдеще време.* Tables 3–15.

Present tense is a simple finite verb form that marks person and number of the subject. Present forms of perfective verbs have a future meaning; however, we prefer morphology (form) to semantics (function) and annotate them `Tense=Pres`, regardless the aspect and meaning.[7]

Future tense of imperfective verbs is usually formed periphrastically, using infinitive or participle of the content verb, and special forms of the auxiliary verb *to be*, e.g. [cs] *budu dělat* "I will do". These special forms are different from the present forms and they are annotated `Tense=Fut`. The infinitive of the content verb does not have the tense feature.

In Croatian, the periphrastic future is formed using another auxiliary verb, *htjeti* "will / want". This verb can also be used as a content (non-auxiliary) verb, and its auxiliary forms are not different from its normal present forms. Therefore they will be annotated `Tense=Pres`.

[7]Some tagsets prefer to call these forms *non-past verb*, cf. Przepiórkowski and Woliński (2003).

Nu	Sing			Dual			Plur		
Pe	1	2	3	1	2	3	1	2	3
cs	*budu*	*budeš*	*bude*				*budeme*	*budete*	*budou*
sk	*budem*	*budeš*	*bude*				*budeme*	*budete*	*budú*
hsb	*budu*	*budźeš*	*budźe*	*budźemoj*	*budźetej*	*budźetej*	*budźemy*	*budźeće*	*budu*
pl	*będę*	*będziesz*	*będzie*				*będziemy*	*będziecie*	*będą*
uk	*буду* *budu*	*будеш* *budeš*	*буде* *bude*				*будемо* *budemo*	*будете* *budete*	*будуть* *budut′*
ru	*буду* *budu*	*будешь* *budeš′*	*будет* *budet*				*будем* *budem*	*будете* *budete*	*будут* *budut*
sl	*bom*	*boš*	*bo*	*bova*	*bosta*	*bosta*	*bomo*	*boste*	*bodo*
cu	бѫдѫ *bǫdǫ*	бѫдеши *bǫdeši*	бѫдетъ *bǫdetъ*	бѫдевѣ *bǫdevě*	бѫдета *bǫdeta*	бѫдете *bǫdete*	бѫдемъ *bǫdemъ*	бѫдете *bǫdete*	бѫдѫтъ *bǫdǫtъ*

Table 4. To be, `VerbForm=Fin | Mood=Ind | Tense=Fut`.

A handful of Czech, Slovak and Slovenian motion verbs also have simple future forms, created by the prefix *p[oóôů]-*: [cs] *půjde* “he will go”, *pojede* “he will ride”, *poletí* “he will fly” but also *pokvete* “it will bloom”. In these cases the prefix is not derivational because it does not create a new perfective lemma with a full paradigm. Thus we annotate these forms as future so they are distinguished from the present forms. In other languages the situation may be different. Russian *пойти (pojti)* is a full perfective counterpart of the imperfective *идти (idti)* and its present forms are annotated `Tense=Pres`.

Ukrainian is special in that it has regular simple future forms of imperfective verbs (not restricted to motion verbs). The periphrastic future also exists.

Number	Sing			Dual		
Person	1	2	3	1	2	3
cs	*půjdu*	*půjdeš*	*půjde*			
sk	*pôjdem*	*pôjdeš*	*pôjde*			
hsb	*póńdu*	*póńdźeš*	*póńdźe*	*póńdźemoj*	*póńdźetej*	*póńdźetej*
sl	*pojdem*	*pojdeš*	*pojde*	*pojdeva*	*pojdeta*	*pojdeta*
uk	*йтиму* *jtymu*	*йтимеш* *jtymeš*	*йтиме* *jtyme*			

…

…

Number	Plur		
Person	1	2	3
cs	*půjdeme*	*půjdete*	*půjdou*
sk	*pôjdeme*	*pôjdete*	*pôjdu*
hsb	*póńdźemy*	*póńdźeće*	*póńdu*
sl	*pojdemo*	*pojdete*	*pojdejo*
uk	*йтимемо* *jtymemo*	*йтимете* *jtymete*	*йтимуть* *jtymut′*

Table 5. To go, `VerbForm=Fin | Mood=Ind | Tense=Fut`*.*

Number	Person	be	can	go	do	accept
Sing	1	*jsem*	*můžu, mohu*	*jdu*	*dělám*	*akceptuji*
Sing	2	*jsi*	*můžeš*	*jdeš*	*děláš*	*akceptuješ*
Sing	3	*je*	*může*	*jde*	*dělá*	*akceptuje*
Plur	1	*jsme*	*můžeme*	*jdeme*	*děláme*	*akceptujeme*
Plur	2	*jste*	*můžete*	*jdete*	*děláte*	*akceptujete*
Plur	3	*jsou*	*můžou, mohou*	*jdou*	*dělají*	*akceptují*

Table 6. [cs] `VerbForm=Fin | Mood=Ind | Tense=Pres`

Number	Person	be	can	go	do	accept
Sing	1	*som*	*môžem*	*idu*	*robím*	*akceptujem*
Sing	2	*si*	*môžeš*	*ideš*	*robíš*	*akceptuješ*
Sing	3	*je*	*môže*	*ide*	*robí*	*akceptuje*
Plur	1	*sme*	*môžeme*	*ideme*	*robíme*	*akceptujeme*
Plur	2	*ste*	*môžete*	*idete*	*robíte*	*akceptujete*
Plur	3	*sú*	*môžu*	*idú*	*robia*	*akceptujú*

Table 7. [sk] VerbForm=Fin | Mood=Ind | Tense=Pres

Number	Person	be	can	go	do	accept
Sing	1	*sym*	*móžu*	*du*	*dźěłam*	*akceptuju*
Sing	2	*sy*	*móžeš*	*dźeš*	*dźěłaš*	*akceptuješ*
Sing	3	*je*	*móže*	*dźe*	*dźěła*	*akceptuje*
Dual	1	*smój*	*móžemoj*	*dźemoj*	*dźěłamoj*	*akceptujemoj*
Dual	2	*staj*	*móžetej*	*dźetej*	*dźěłatej*	*akceptujetej*
Dual	3	*staj*	*móžetej*	*dźetej*	*dźěłatej*	*akceptujetej*
Plur	1	*smy*	*móžemy*	*dźemy*	*dźěłamy*	*akceptujemy*
Plur	2	*sće*	*móžeće*	*dźeće*	*dźěłaće*	*akceptujeće*
Plur	3	*su*	*móža, móžeja*	*du, dźeja*	*dźěłaja*	*akceptuja*

Table 8. [hsb] VerbForm=Fin | Mood=Ind | Tense=Pres

Number	Person	be	can	go	do	accept
Sing	1	*jestem*	*mogę*	*idę*	*robię*	*akceptuję*
Sing	2	*jesteś*	*możesz*	*idziesz*	*robisz*	*akceptujesz*
Sing	3	*jest*	*może*	*idzie*	*robi*	*akceptuje*
Plur	1	*jesteśmy*	*możemy*	*idziemy*	*robimy*	*akceptujemy*
Plur	2	*jesteście*	*możecie*	*idziecie*	*robicie*	*akceptujecie*
Plur	3	*są*	*mogą*	*idą*	*robią*	*akceptują*

Table 9. [pl] VerbForm=Fin | Mood=Ind | Tense=Pres

Number	Person	be	can	go	do	accept
Sing	1	*є* *je*	*можу* *možu*	*йду* *jdu*	*роблю* *roblju*	*акцептую* *akceptuju*
Sing	2	*єси, є* *jesy, je*	*можеш* *možeš*	*йдеш* *jdeš*	*робиш* *robyš*	*акцептуєш* *akceptuješ*
Sing	3	*є* *je*	*може* *može*	*йде* *jde*	*робить* *robyt'*	*акцептує* *akceptuje*
Plur	1	*є* *je*	*можемо* *možemo*	*йдемо, йдем* *jdemo, jdem*	*робимо, робим* *robymo, robym*	*акцептуємо* *akceptujemo*
Plur	2	*є* *je*	*можете* *možete*	*йдете* *jdete*	*робите* *robyte*	*акцептуєте* *akceptujete*
Plur	3	*є* *je*	*можуть* *možut'*	*йдуть* *jdut'*	*роблять* *robljat'*	*акцептують* *akceptujut'*

Table 10. [uk] VerbForm=Fin | Mood=Ind | Tense=Pres

Number	Person	be	can	go	do	accept
Sing	1		*могу* *mogu*	*иду* *idu*	*делаю* *delaju*	*акцептую* *akceptuju*
Sing	2		*можешь* *možeš'*	*идёшь* *idëš'*	*делаешь* *delaeš'*	*акцептуешь* *akceptueš'*
Sing	3	*есть* *est'*	*может* *možet*	*идёт* *idët*	*делает* *delaet*	*акцептует* *akceptuet*
Plur	1		*можем* *možem*	*идём* *idëm*	*делаем* *delaem*	*акцептуем* *akceptuem*
Plur	2		*можете* *možete*	*идёте* *idëte*	*делаете* *delaete*	*акцептуете* *akceptuete*
Plur	3	*суть* *sut'*	*могут* *mogut*	*идут* *idut*	*делают* *delajut*	*акцептуют* *akceptujut*

Table 11. [ru] VerbForm=Fin | Mood=Ind | Tense=Pres

Number	Person	be	can	go	do	accept
Sing	1	*sem*	*morem*	*grem*	*delam*	*akceptiram*
Sing	2	*si*	*moreš*	*greš*	*delaš*	*akceptiraš*
Sing	3	*je*	*more*	*gre*	*dela*	*akceptira*
Dual	1	*sva*	*moreva*	*greva*	*delava*	*akceptirava*
Dual	2	*sta*	*moreta*	*gresta*	*delata*	*akceptirata*
Dual	3	*sta*	*moreta*	*gresta*	*delata*	*akceptirata*
Plur	1	*smo*	*moremo*	*gremo*	*delamo*	*akceptiramo*
Plur	2	*ste*	*morete*	*greste*	*delate*	*akceptirate*
Plur	3	*so*	*morejo*	*gredo, grejo*	*delajo*	*akceptirajo*

Table 12. [sl] VerbForm=Fin | Mood=Ind | Tense=Pres

Number	Person	be	can	go	do	accept
Sing	1	*съм* *săm*	*мога* *moga*	*отивам* *otivam*	*правя* *pravja*	*акцептирам* *akceptiram*
Sing	2	*си* *si*	*можеш* *možeš*	*отиваш* *otivaš*	*правиш* *praviš*	*акцептираш* *akceptiraš*
Sing	3	*е* *e*	*може* *može*	*отива* *otiva*	*прави* *pravi*	*акцептира* *akceptira*
Plur	1	*сме* *sme*	*можем* *možem*	*отиваме* *otivame*	*правим* *pravim*	*акцептираме* *akceptirame*
Plur	2	*сте* *ste*	*можете* *možete*	*отивате* *otivate*	*правите* *pravite*	*акцептирате* *akceptirate*
Plur	3	*са* *sa*	*могат* *mogat*	*отиват* *otivat*	*правят* *pravjat*	*акцептират* *akceptirat*

Table 13. [bg] VerbForm=Fin | Mood=Ind | Tense=Pres

Number	Person	be	can	go	do	accept
Sing	1	*jesam, sam*	*mogu*	*idem*	*delam*	*akceptiram*
Sing	2	*jesi, si*	*možeš*	*ideš*	*delaš*	*akceptiraš*
Sing	3	*jest, je*	*može*	*ide*	*dela*	*akceptira*
Plur	1	*jesmo, smo*	*možemo*	*idemo*	*delamo*	*akceptiramo*
Plur	2	*jeste, ste*	*možete*	*idete*	*delate*	*akceptirate*
Plur	3	*jesu, su*	*mogu*	*idu*	*delaju*	*akceptiraju*

Table 14. [hr] `VerbForm=Fin | Mood=Ind | Tense=Pres`

Number	Person	be	can	go	do
Sing	1	есмь *jesmь*	могѫ *mogǫ*	идѫ, ідѫ *idǫ*	дѣлаѭ *dělajǫ*
Sing	2	еси *jesi*	можеши *možeši*	идеши, ідеши *ideši*	дѣлаеши *dělaješi*
Sing	3	естъ *jestъ*	можетъ *možetъ*	идетъ, ідетъ *idetъ*	дѣлаатъ *dělaatъ*
Dual	1	есвѣ *jesvě*	можевѣ *moževě*	идевѣ, ідевѣ *idevě*	дѣлаевѣ *dělajevě*
Dual	2	еста *jesta*	можета *možeta*	идета, ідета *ideta*	дѣлаета *dělajeta*
Dual	3	есте *jeste*	можете *možete*	идете, ідете *idete*	дѣлаете *dělajete*
Plur	1	есмъ *jesmъ*	можемъ *možemъ*	идемъ, ідемъ *idemъ*	дѣлаемъ *dělajemъ*
Plur	2	есте *jeste*	можете *možete*	идете, ідете *idete*	дѣлаете *dělajete*
Plur	3	сѫтъ *sǫtъ*	могѫтъ *mogǫtъ*	идѫтъ, ідѫтъ *idǫtъ*	дѣлаѭтъ *dělajǫtъ*

Table 15. [cu] `VerbForm=Fin | Mood=Ind | Tense=Pres`

7. Imperative

[cs] *rozkazovací způsob (imperativ);* [sk] *imperatív (rozkazovací spôsob);* [hsb] *imperatiw;* [pl] *tryb rozkazujący;* [uk] *наказовий спосіб;* [ru] *повелительное наклонение;* [sl] *velelnik, velelni naklon;* [hr] *imperativ;* [bg] *повелително наклонение (императив).* Tables 16–25.

Imperative is a simple finite verb form that marks person and number but it does not mark tense (we leave the `Tense` feature empty). Imperative forms are not available in the 3rd person (appeals to third persons may be formed periphrastically, using particles and present indicative forms; these are not annotated as imperatives). Imperative also does not exist in the 1st person singular. Modal verbs usually do not have imperatives.

`Number`	`Person`	be	go	do	accept
`Sing`	`2`	*buď*	*jdi, pojď*	*dělej*	*akceptuj*
`Plur`	`1`	*buďme*	*jděme, pojďme*	*dělejme*	*akceptujme*
`Plur`	`2`	*buďte*	*jděte, pojďte*	*dělejte*	*akceptujte*

Table 16. [cs] `VerbForm=Fin | Mood=Imp`

`Number`	`Person`	be	go	do	accept
`Sing`	`2`	*buď*	*choď*	*rob*	*akceptuj*
`Plur`	`1`	*buďme*	*choďme*	*robme*	*akceptujme*
`Plur`	`2`	*buďte*	*choďte*	*robte*	*akceptujte*

Table 17. [sk] `VerbForm=Fin | Mood=Imp`

`Number`	`Person`	be	go	do	accept
`Sing`	`2`	*budź*	*dźi, póńdź*	*dźěłaj*	*akceptuj*
`Dual`	`1`	*budźmoj*	*dźemoj, póńdźmoj*	*dźěłajmoj*	*akceptujmoj*
`Dual`	`2`	*budźtej*	*dźetej, póńdźtej*	*dźěłajtej*	*akceptujtej*
`Plur`	`1`	*budźmy*	*dźemy, póńdźmy*	*dźěłajmy*	*akceptujmy*
`Plur`	`2`	*budźće*	*dźeće, póńdźće*	*dźěłajće*	*akceptujće*

Table 18. [hsb] `VerbForm=Fin | Mood=Imp`

Number	Person	be	go	do	accept
Sing	2	*bądź*	*idź*	*rób*	*akceptuj*
Plur	1	*bądźmy*	*idźmy*	*róbmy*	*akceptujmy*
Plur	2	*bądźcie*	*idźcie*	*róbcie*	*akceptujcie*

Table 19. [pl] VerbForm=Fin | Mood=Imp

Number	Person	be	go	do	accept
Sing	2	*будь* *bud'*	*йди* *jdy*	*роби* *roby*	*акцептуй* *akceptuj*
Plur	1	*будьмо* *bud'mo*	*йдімо, йдім* *jdimo, jdim*	*робімо, робім* *robimo, robim*	*акцептуймо* *akceptujmo*
Plur	2	*будьте* *bud'te*	*йдіть* *jdit'*	*робіть* *robit'*	*акцептуйте* *akceptujte*

Table 20. [uk] VerbForm=Fin | Mood=Imp

Number	Person	be	go	do	accept
Sing	2	*будь* *bud'*	*иди* *idi*	*делай* *delaj*	*акцептуй* *akceptuj*
Plur	1	*будемте* *budemte*	*идёмте* *idëmte*	*делаемте* *delaemte*	*акцептуемте* *akceptuemte*
Plur	2	*будьте* *bud'te*	*идите* *idite*	*делайте* *delajte*	*акцептуйте* *akceptujte*

Table 21. [ru] VerbForm=Fin | Mood=Imp

Number	Person	be	go	do	accept
Sing	2	*bodi*	*pojdi*	*delaj*	*akceptiraj*
Dual	1	*bodiva*	*pojdiva*	*delajva*	*akceptirajva*
Dual	2	*bodita*	*pojdita*	*delajta*	*akceptirajta*
Plur	1	*bodimo*	*pojdimo*	*delajmo*	*akceptirajmo*
Plur	2	*bodite*	*pojdite*	*delajte*	*akceptirajte*

Table 22. [sl] VerbForm=Fin | Mood=Imp

Number	Person	be	go	do	accept
Sing	2	*budi*	*idi*	*delaj*	*akceptiraj*
Plur	1	*budimo*	*idimo*	*delajmo*	*akceptirajmo*
Plur	2	*budite*	*idite*	*delajte*	*akceptirajte*

Table 23. [hr] VerbForm=Fin | Mood=Imp

Number	Person	be	go	do	accept
Sing	2	*бъди* *bădi*	*отивай* *otivaj*	*прави* *pravi*	*акцептирай* *akceptiraj*
Plur	2	*бъдете* *bădete*	*отивайте* *otivajte*	*правете* *pravete*	*акцептирайте* *akceptirajte*

Table 24. [bg] VerbForm=Fin | Mood=Imp

Number	Person	be	go	do
Sing	2	бѫди *bǫdi*	иди, іди *idi*	дѣлаи *dělai*
Dual	2	бѫдѣта *bǫděta*	идѣта, ідѣта *iděta*	дѣлаита *dělaita*
Plur	2	бѫдѣте *bǫděte*	идѣте, ідѣте *iděte*	дѣлаите *dělaite*

Table 25. [cu] VerbForm=Fin | Mood=Imp

8. Aorist Indicative

[cs] *aorist;* [hsb] *preteritum;* [hr] *aorist (pređašnje svršeno vreme);* [bg] *минало свършено време.* Tables 26, 32, 28 and 30.

Aorist is the old Slavic simple past tense. It is a finite form that marks person and number of the subject. It existed in the Old Church Slavonic language and it has survived in several languages until today; however, many languages have replaced it by the l-participle. For example, aorist is attested in Old Czech but it vanished during the 15th century.

Aorist is regularly used (together with imperfect, see Section 9) in Bulgarian and Macedonian. It is still understood in Serbian and Croatian, albeit its usage is limited. Aorist has also survived in the Sorbian languages, where it has effectively merged with imperfect into one simple past called preterite. Unlike in Bulgarian, in Sorbian the forms stemming from aorist are only found with perfective verbs, and the historical forms of imperfect only with imperfective verbs[8] (Breu, 2000). Hence we have just two inflection patterns, instead of two different tenses.

We can use the simple `Tense=Past` feature to annotate aorist in Slavic languages as it does not collide with the other past forms. This has been the original intention in Interset and in Universal Dependencies and it is used currently both in the Old Church Slavonic and the Bulgarian data. On the other hand, UD Ancient Greek uses a language-specific value `Tense=Aor`; if the future versions of the *universal* guidelines adopt this value, it might be more appropriate to use it.

The Sorbian preterite will be also tagged `Tense=Past`, regardless whether the verb is perfective or imperfective.

9. Imperfect Indicative

[cs] *imperfektum;* [hr] *imperfekat (pređašnje nesvršeno vreme);* [bg] *минало несвършено време.* Tables 27, 29 and 31.

Imperfect is another simple past tense that only survived in a few languages. It does not have any equivalent in English, but there are imperfect tenses in Romance languages.

For the merged aorist-imperfect (preterite) in Sorbian languages, see Section 8.

Verbs in imperfect describe states or actions that were happening during some past moment. They may or may not continue at and after the moment of speaking. Important is the past context and the relation of the action (state) to some other action (state) happening in the past.

Despite the name, both imperfective and perfective verbs can be used in the imperfect tense! Perfective verbs in the imperfect tense denote actions that were repeated in

[8] It could be argued that the Sorbian usage is prototypical, while the imperfect tense of perfective verbs in Bulgarian is marked. Nevertheless, such change of perspective would have no impact on our proposed analysis.

Number	Person	be	can	go	do	accept
Sing	1	*bych*	*možech*	*jid*	*dělach*	*přijiech*
Sing	2	*by*	*može*	*jide*	*děla*	*přijie*
Sing	3	*by*	*može*	*jide*	*děla*	*přijie*
Dual	1	*bychově*	*možechově*	*jidově*	*dělachově*	*přijiechově*
Dual	2	*bysta*	*možesta*	*jideta*	*dělasta*	*přijiesta*
Dual	3	*bysta*	*možesta*	*jideta*	*dělasta*	*přijiesta*
Plur	1	*bychom*	*možechom*	*jidom*	*dělachom*	*přijiechom*
Plur	2	*byste*	*možeste*	*jidete*	*dělaste*	*přijieste*
Plur	3	*bychu*	*možechu*	*jidú*	*dělachu*	*přijiechu*

Table 26. Old [cs] `VerbForm=Fin | Mood=Ind | Tense=Past`

Number	Person	be	can	go	do	accept
Sing	1	*biech*	*možiech*	*jdiech*	*dělajiech*	*přijiech*
Sing	2	*bieše*	*možieše*	*jdieše*	*dělajieše*	*přijieše*
Sing	3	*bieše*	*možieše*	*jdieše*	*dělajieše*	*přijieše*
Dual	1	*biechově*	*možiechově*	*jdiechově*	*dělajiechově*	*přijiechově*
Dual	2	*biešta*	*možiešta*	*jdiešta*	*dělajiešta*	*přijiešta*
Dual	3	*biešta*	*možiešta*	*jdiešta*	*dělajiešta*	*přijiešta*
Plur	1	*biechom*	*možiechom*	*jdiechom*	*dělajiechom*	*přijiechom*
Plur	2	*biešte*	*možiešte*	*jdiešte*	*dělajiešte*	*přijiešte*
Plur	3	*biechu*	*možiechu*	*jdiechu*	*dělajiechu*	*přijiechu*

Table 27. Old [cs] `VerbForm=Fin | Mood=Ind | Tense=Imp`

the past. Hence the `Aspect` feature should *not* be used to mark this tense. As discussed in Section 4, that feature should be reserved to denote the lexical aspect of Slavic verbs, bound to their lemma. Instead, Universal Features provide a feature dedicated to the imperfect tense, `Tense=Imp`.

Examples [bg]:

- *Когато се прибрах вкъщи, децата вече* ***спяха****. (Kogato se pribrah vkăšti, decata veče spjaha)* “When I came home, the children were already asleep.”
- *Щом* ***дойдеше****, веднага запалваше цигара. (Štom dojdeše, vednaga zapalvaše cigara.)* “Every time he came, he always lit a cigarette.”

Number	Person	be	can	go	do	accept
Sing	1	*бях* *bjah*	*можах* *možah*	*отивах* *otivah*	*правих* *pravih*	*акцептирах* *akceptirah*
Sing	2	*беше, бе* *beše, be*	*можа* *moža*	*отива* *otiva*	*прави* *pravi*	*акцептира* *akceptira*
Sing	3	*беше, бе* *beše, be*	*можа* *moža*	*отива* *otiva*	*прави* *pravi*	*акцептира* *akceptira*
Plur	1	*бяхме* *bjahme*	*можахме* *možahme*	*отивахме* *otivahme*	*правихме* *pravihme*	*акцептирахме* *akceptirahme*
Plur	2	*бяхте* *bjahte*	*можахте* *možahte*	*отивахте* *otivahte*	*правихте* *pravihte*	*акцептирахте* *akceptirahte*
Plur	3	*бяха* *bjaha*	*можаха* *možaha*	*отиваха* *otivaha*	*правиха* *praviha*	*акцептираха* *akceptiraha*

Table 28. [bg] VerbForm=Fin | Mood=Ind | Tense=Past

Number	Person	be	can	go	do	accept
Sing	1	*бях* *bjah*	*можех* *možeh*	*отивах* *otivah*	*правех* *praveh*	*акцептирах* *akceptirah*
Sing	2	*беше, бе* *beše, be*	*можеше* *možeše*	*отиваше* *otivaše*	*правеше* *praveše*	*акцептираше* *akceptiraše*
Sing	3	*беше, бе* *beše, be*	*можеше* *možeše*	*отиваше* *otivaše*	*правеше* *praveše*	*акцептираше* *akceptiraše*
Plur	1	*бяхме* *bjahme*	*можехме* *možehme*	*отивахме* *otivahme*	*правехме* *pravehme*	*акцептирахме* *akceptirahme*
Plur	2	*бяхте* *bjahte*	*можехте* *možehte*	*отивахте* *otivahte*	*правехте* *pravehte*	*акцептирахте* *akceptirahte*
Plur	3	*бяха* *bjaha*	*можеха* *možeha*	*отиваха* *otivaha*	*правеха* *praveha*	*акцептираха* *akceptiraha*

Table 29. [bg] VerbForm=Fin | Mood=Ind | Tense=Imp

Number	Person	be	can	go	do
Sing	1	быхъ *bychъ*	могъ *mogъ*	идъ, ꙇдъ *idъ*	дѣлахъ *dělachъ*
Sing	2	быстъ *bystъ*	може *može*	иде, ꙇде *ide*	дѣлаше *dělaše*
Sing	3	быстъ, быⷭ҇ *bystъ, by*	може *može*	иде, ꙇде *ide*	дѣлаше *dělaše*
Dual	1	быховѣ *bychově*	моговѣ *mogově*	идовѣ, ꙇдовѣ *idově*	дѣлаховѣ *dělachově*
Dual	2	быста *bysta*	можета *možeta*	идета, ꙇдета *ideta*	дѣласта *dělasta*
Dual	3	бысте *byste*	можете *možete*	идете, ꙇдете *idete*	дѣласте *dělaste*
Plur	1	быхомъ *bychomъ*	могомъ *mogomъ*	идомъ, ꙇдомъ *idomъ*	дѣлахомъ *dělachomъ*
Plur	2	бысте *byste*	можете *možete*	идете, ꙇдете *idete*	дѣласте *dělaste*
Plur	3	бышѧ *byšę*	могѫ *mogǫ*	идѫ, ꙇдѫ *idǫ*	дѣлашѧ *dělašę*

Table 30. [cu] `VerbForm=Fin | Mood=Ind | Tense=Past`

Numb	P	be	can	go	do
Sing	1	бѣхъ *běchъ*	можаахъ *možaachъ*	идѣахъ, ідѣахъ *iděachъ*	дѣлаахъ *dělaachъ*
Sing	2	бѣ *bě*	можааше *možaaše*	идѣаше, ідѣаше *iděaše*	дѣлааше *dělaaše*
Sing	3	бѣ, бѣаше *bě, běaše*	можааше *možaaše*	идѣаше, ідѣаше *iděaše*	дѣлааше *dělaaše*
Dual	1	бѣховѣ *běchově*	можааховѣ *možaachově*	идѣаховѣ, ідѣаховѣ *iděachově*	дѣлааховѣ *dělaachově*
Dual	2	бѣста *běsta*	можаашета *možaašeta*	идѣашета, ідѣашета *iděašeta*	дѣлаашета *dělaašeta*
Dual	3	бѣашете, бѣсте *běašete, běste*	можаашете *možaašete*	идѣашете, ідѣашете *iděašete*	дѣлаашете *dělaašete*
Plur	1	бѣхомъ *běchomъ*	можаахомъ *možaachomъ*	идѣахомъ, ідѣахомъ *iděachomъ*	дѣлаахомъ *dělaachomъ*
Plur	2	бѣсте *běste*	можаашете *možaašete*	идѣашете, ідѣашете *iděašete*	дѣлаашете *dělaašete*
Plur	3	бѣахѫ, бѣшѧ *běachǫ, běšę*	можаахѫ *možaachǫ*	идѣахѫ, ідѣахѫ *iděachǫ*	дѣлаахѫ *dělaachǫ*

Table 31. [cu] VerbForm=Fin | Mood=Ind | Tense=Imp

Number	Person	be	can	go	do	accept
Sing	1	*běch*	*móžech*	*dźěch*	*dźěłach*	*akceptowach*
Sing	2	*běše*	*móžeše*	*dźěše*	*dźěłaše*	*akceptowaše*
Sing	3	*běše*	*móžeše*	*dźěše*	*dźěłaše*	*akceptowaše*
Dual	1	*běchmoj*	*móžechmoj*	*dźěmoj*	*dźěłachmoj*	*akceptowachmoj*
Dual	2	*běštej*	*móžeštej*	*dźěštej*	*dźěłaštej*	*akceptowaštej*
Dual	3	*běštej*	*móžeštej*	*dźěštej*	*dźěłaštej*	*akceptowaštej*
Plur	1	*běchmy*	*móžechmy*	*dźěchmy*	*dźěłachmy*	*akceptowachmy*
Plur	2	*běšće*	*móžešće*	*dźěšće*	*dźěłašće*	*akceptowašće*
Plur	3	*běchu*	*móžechu*	*dźěchu*	*dźěłachu*	*akceptowachu*

Table 32. [hsb] VerbForm=Fin | Mood=Ind | Tense=Past

10. Active Participle and Past Tense

[cs] *příčestí činné, minulý čas;* [sk] *minulý čas;* [hsb] *ł-forma, perfekt;* [pl] *czas przeszły;* [uk] *минулий час;* [ru] *прошедшее время;* [sl] *opisni deležnik na -l, preteklik;* [hr] *glagolski pridjev radni, prošlo vreme;* [bg] *минало деятелно свършено причастие, минало деятелно несвършено причастие.* Tables 33–42.

The typical formation of the past tense in most (but not all) modern Slavic languages is periphrastic, using a finite form of the auxiliary verb *to be* and the active participle (as opposed to the passive participle). The participle may also be called past participle because of its close ties to the past tense, and despite the fact that it is also used to form conditional or even the future tense. Sometimes the participle itself is called past tense (it makes sense because in some languages the auxiliary verb is omitted). Or it is simply called l-participle because its suffixes typically involve the consonant *-l*.

Early stages of Slavic languages (and those modern stages that retained the aorist) understand the constructions with the l-participle as perfect tenses that we know in English. Present perfect, past perfect and future perfect may be constructed, depending on the form of the auxiliary verb. Interestingly, the periphrastic past tense is also termed *préteritum* in Modern Czech (Academia, 1986), but the term *perfektum* prevails when Old Czech is described (Komárek et al., 1967) (cf. *Präterium = Imperfekt* vs. *Perfekt* in German).

Like other Slavic participles, the l-participle marks gender and number. Typically it has only the short form that is used in predicates, it does not inflect for case and is tagged `VERB` or `AUX`. Occasional long forms exist but they are considered derived adjectives and tagged `ADJ`. The derivation is not productive. It applies mainly to intransitive perfective verbs, while the passive participle would be used with transitive verbs for the same purpose. Example [cs]: *spadlý* "the one who fell down", *shnilý* "rotten", *pokleslý* "dropped". Annotating `VerbForm` of the derived adjective is purely optional. The short, predicative form should always have `VerbForm=Part`.

`Voice=Act` should also be always present so that the participle is distinguished from the passive participle.

Some Bulgarian verbs have two l-participles (past participles): perfect and imperfect. We cannot use the `Aspect` feature to distinguish them because the feature is bound to lemma, and an imperfective verb can have both perfect and imperfect participles. Nevertheless, the distinction is an analogy to the distinction between the two simple past tenses, and we will use the `Tense` feature to distinguish the participles. The default is `Tense=Past` (for past perfect participles). Past imperfect participles will get `Tense=Imp`.

It is less clear whether the l-participle should be annotated with `Tense=Past` in the other languages, in which it is not necessary to distinguish different types of l-participles. In many Slavic languages (especially the northern ones) this is the promi-

<table>
<tr><th>Number</th><th>Gender</th><th>Animacy</th><th>be</th><th>can</th><th>go</th><th>do</th><th>accept</th></tr>
<tr><td>Sing</td><td>Masc</td><td></td><td>byl</td><td>mohl</td><td>šel</td><td>dělal</td><td>akceptoval</td></tr>
<tr><td>Sing</td><td>Fem</td><td></td><td>byla</td><td>mohla</td><td>šla</td><td>dělala</td><td>akceptovala</td></tr>
<tr><td>Sing</td><td>Neut</td><td></td><td>bylo</td><td>mohlo</td><td>šlo</td><td>dělalo</td><td>akceptovalo</td></tr>
<tr><td>Plur</td><td>Masc</td><td>Anim</td><td>byli</td><td>mohli</td><td>šli</td><td>dělali</td><td>akceptovali</td></tr>
<tr><td>Plur</td><td>Masc</td><td>Inan</td><td rowspan="2">byly</td><td rowspan="2">mohly</td><td rowspan="2">šly</td><td rowspan="2">dělaly</td><td rowspan="2">akceptovaly</td></tr>
<tr><td>Plur</td><td>Fem</td><td></td></tr>
<tr><td>Plur</td><td>Neut</td><td></td><td>byla</td><td>mohla</td><td>šla</td><td>dělala</td><td>akceptovala</td></tr>
</table>

Table 33. [cs] VERB,AUX | VerbForm=Part | Voice=Act | Tense=Past

Number	Gender	be	can	go	do	accept
Sing	Masc	*bol*	*mohol*	*išiel*	*robil*	*akceptoval*
Sing	Fem	*bola*	*mohla*	*išla*	*robila*	*akceptovala*
Sing	Neut	*bolo*	*mohlo*	*išlo*	*robilo*	*akceptovalo*
Plur		*boli*	*mohli*	*išli*	*robili*	*akceptovali*

Table 34. [sk] VERB,AUX | VerbForm=Part | Voice=Act | Tense=Past

nent and default function of the l-participle.[9] Even in languages where it is used in periphrastic perfect tenses (which co-exist with simple past tenses), the perfect or resultative meaning implies that the action happened in the past, although the past is relative to a point in time that may be different from the moment of speaking. Therefore we recommend to include `Tense=Past` in the annotation.

See Section 18 for the annotation of l-participles used in the current UD datasets.

In Slovenian and Serbo-Croatian, the finite form of the auxiliary is used with all persons and numbers: *Je šel v šolo.* "He went to the school." *Sem šel v šolo.* "I went to the school." In Czech and Slovak, the finite form of the auxiliary is omitted in the 3rd person: *Šel do školy.* "He went to the school." *Šel jsem do školy.* "I went to the school." In Ukrainian and Russian, the auxiliary is omitted in all persons. That is why the subject cannot be dropped in Russian. The person could be understood from a finite verb but not from the participle, hence we need a personal pronoun: *Он пошел*

[9] As mentioned above, it is also used in conditional and in some languages even in the future tense. Still, we are looking for distinctive features of individual words rather than of the periphrastic expressions. In a Slavic-wide perspective, `Past` seems as close as we can get without defining a language-specific feature for l-participles.

Number	Gender	be	can	go	do	accept
Sing	Masc	*był*	*mohł*	*šoł*	*dźěłał*	*akceptował*
Sing	Fem	*była*	*móhła*	*šła*	*dźěłała*	*akceptowała*
Sing	Neut	*było*	*móhło*	*šło*	*dźěłało*	*akceptowało*
Dual		*byłoj*	*móhłoj*	*šłoj*	*dźěłałoj*	*akceptowałoj*
Plur		*byli*	*móhli*	*šli*	*dźěłali*	*akceptowali*

Table 35. [hsb] VERB,AUX | VerbForm=Part | Voice=Act | Tense=Past

<table>
<tr><th>Number</th><th>Gender</th><th>Animacy</th><th>be</th><th>can</th><th>go</th><th>do</th><th>accept</th></tr>
<tr><td>Sing</td><td>Masc</td><td></td><td>był</td><td>mógł</td><td>szedł</td><td>robił</td><td>akceptował</td></tr>
<tr><td>Sing</td><td>Fem</td><td></td><td>była</td><td>mogła</td><td>szła</td><td>robiła</td><td>akceptowała</td></tr>
<tr><td>Sing</td><td>Neut</td><td></td><td>było</td><td>mogło</td><td>szło</td><td>robiło</td><td>akceptowało</td></tr>
<tr><td>Plur</td><td>Masc</td><td>Anim</td><td>byli</td><td>mogli</td><td>szli</td><td>robili</td><td>akceptowali</td></tr>
<tr><td>Plur</td><td>Masc</td><td>Nhum</td><td rowspan="4">były</td><td rowspan="4">mogły</td><td rowspan="4">szły</td><td rowspan="4">robiły</td><td rowspan="4">akceptowały</td></tr>
<tr><td>Plur</td><td>Masc</td><td>Inan</td></tr>
<tr><td>Plur</td><td>Fem</td><td></td></tr>
<tr><td>Plur</td><td>Neut</td><td></td></tr>
</table>

Table 36. [pl] VERB,AUX | VerbForm=Part | Voice=Act | Tense=Past

в школу. (On pošel v školu.) "He went to the school." *Я пошел в школу. (Ja pošel v školu.)* "I went to the school."

In Polish, the auxiliary and the participle have merged in one past-tense form. However, they can also attach to a preceding word: *Cieszę się, że**ś** zrozumiał.* "I am glad that you have understood." (The auxiliary *-ś* is attached to a conjunction.) *My**śmy** nie wiedzieli, że przyjadą.* "We did not know they were coming." (Attached to a pronoun.) That is why the tokenization in the Polish treebank cuts off the finite morpheme as a separate syntactic word of a special type called "agglutination". We keep this approach to tokenization, emphasizing the parallelism between the Polish data and the other Slavic languages: *Poszedł do szkoły.* "He went to the school." *Poszedł-em do szkoły.* "I went to the school." (The hyphen in the second example indicates tokenization but it does not appear in the surface text.)

Note that there are other types of participles that could be (and sometimes are) called active participles. See Section 13 for details.

Number	Gender	be	can	go	do	accept
Sing	Masc	*був* *buv*	*міг* *mih*	*йшов* *jšov*	*робив* *robyv*	*акцептував* *akceptuvav*
Sing	Fem	*була* *bula*	*могла* *mohla*	*йшла* *jšla*	*робила* *robyla*	*акцептувала* *akceptuvala*
Sing	Neut	*було* *bulo*	*могло* *mohlo*	*йшло* *jšlo*	*робило* *robylo*	*акцептувало* *akceptuvalo*
Plur		*були* *buly*	*могли* *mohly*	*йшли* *jšly*	*робили* *robyly*	*акцептували* *akceptuvaly*

Table 37. [uk] VERB,AUX | VerbForm=Part | Voice=Act | Tense=Past

Number	Gender	be	can	go	do	accept
Sing	Masc	*был* *byl*	*мог* *mog*	*шёл* *šël*	*делал* *delal*	*акцептовал* *akceptoval*
Sing	Fem	*была* *byla*	*могла* *mogla*	*шла* *šla*	*делала* *delala*	*акцептовала* *akceptovala*
Sing	Neut	*было* *bylo*	*могло* *moglo*	*шло* *šlo*	*делало* *delalo*	*акцептовало* *akceptovalo*
Plur		*были* *byli*	*могли* *mogli*	*шли* *šli*	*делали* *delali*	*акцептовали* *akceptovali*

Table 38. [ru] VERB,AUX | VerbForm=Part | Voice=Act | Tense=Past

Number	Gender	be	can	go	do	accept
Sing	Masc	*bio*	*mogao*	*šao*	*delao*	*akceptirao*
Sing	Fem	*bila*	*mogla*	*šla*	*delala*	*akceptirala*
Sing	Neut	*bilo*	*moglo*	*šlo*	*delalo*	*akceptiralo*
Plur	Masc	*bili*	*mogli*	*šli*	*delali*	*akceptirali*
Plur	Fem	*bile*	*mogle*	*šle*	*delale*	*akceptirale*
Plur	Neut	*bila*	*mogla*	*šla*	*delala*	*akceptirala*

Table 39. [hr] VERB,AUX | VerbForm=Part | Voice=Act | Tense=Past

Number	Gender	be	can	go	do	accept
Sing	Masc	*bil*	*mogel*	*šel*	*delal*	*akceptiral*
Sing	Fem	*bila*	*mogla*	*šla*	*delala*	*akceptirala*
Sing	Neut	*bilo*	*moglo*	*šlo*	*delalo*	*akceptiralo*
Dual	Masc	*bila*	*mogla*	*šla*	*delala*	*akceptirala*
Dual	Fem	*bili*	*mogli*	*šli*	*delali*	*akceptirali*
Dual	Neut					
Plur	Masc	*bili*	*mogli*	*šli*	*delali*	*akceptirali*
Plur	Fem	*bile*	*mogle*	*šle*	*delale*	*akceptirale*
Plur	Neut	*bila*	*mogla*	*šla*	*delala*	*akceptirala*

Table 40. [sl] VERB,AUX | VerbForm=Part | Voice=Act | Tense=Past

Tense	Number	Gender	be	can	go	do	accept
Past	Sing	Masc	*бил* *bil*	*могъл* *mogăl*	*отивал* *otival*	*правил* *pravil*	*акцептирал* *akceptiral*
Past	Sing	Fem	*била* *bila*	*могла* *mogla*	*отивала* *otivala*	*правила* *pravila*	*акцептирала* *akceptirala*
Past	Sing	Neut	*било* *bilo*	*могло* *moglo*	*отивало* *otivalo*	*правило* *pravilo*	*акцептирало* *akceptiralo*
Past	Plur		*били* *bili*	*могли* *mogli*	*отивали* *otivali*	*правили* *pravili*	*акцептирали* *akceptirali*
Imp	Sing	Masc		*можел* *možel*		*правел* *pravel*	
Imp	Sing	Fem		*можела* *možela*		*правела* *pravela*	
Imp	Sing	Neut		*можело* *moželo*		*правело* *pravelo*	
Imp	Plur			*можели* *moželi*		*правели* *praveli*	

Table 41. [bg] VERB,AUX | VerbForm=Part | Voice=Act

Number	Gender	be	can	go	do
Sing	Masc	былъ *bylъ*	моглъ *moglъ*	шелъ *šelъ*	дѣлалъ *dělalъ*
Sing	Fem	была *byla*	могла *mogla*	шла *šla*	дѣлала *dělala*
Sing	Neut	было *bylo*	могло *moglo*	шло *šlo*	дѣлало *dělalo*
Dual	Masc	была *byla*	могла *mogla*	шла *šla*	дѣлала *dělala*
Dual	Fem Neut	былѣ *bylě*	моглѣ *moglě*	шлѣ *šlě*	дѣлалѣ *dělalě*
Plur	Masc	были *byli*	могли *mogli*	шли *šli*	дѣлали *dělali*
Plur	Fem	былы *byly*	моглы *mogly*	шлы *šly*	дѣлалы *dělaly*
Plur	Neut	была *byla*	могла *mogla*	шла *šla*	дѣлала *dělala*

Table 42. [cu] VERB,AUX | VerbForm=Part | Voice=Act | Tense=Past

Number	Sing			Dual			Plur		
Person	1	2	3	1	2	3	1	2	3
cs	*bych*	*bys*	*by*				*bychom*	*byste*	*by*
sk	*by*								
hsb	*bych*	*by*	*by*	*bychmoj*	*byštej*	*byštej*	*bychmy*	*byšće*	*bychu*
pl	*-bym*	*-byś*	*-by*				*-byśmy*	*-byście*	*-by*
uk	*б, би* *b, by*								
ru	*бы, б* *by, b*								
sl	*bi*								
hr	*bih*	*bi*	*bi*				*bismo*	*biste*	*bi*
bg	*бих* *bih*	*би* *bi*	*би* *bi*				*бихме* *bihme*	*бихте* *bihte*	*биха* *biha*
cu	бимь *bimь*	би *bi*	би *bi*	бивѣ *bivě*	биста *bista*	бисте *biste*	бимъ *bimъ*	бисте *biste*	бѫ, бишѧ *bǫ, bišę*

Table 43. To be, AUX | VerbForm=Fin | Mood=Cnd.

11. Conditional

[cs] *podmiňovací způsob;* [sk] *podmieňovací spôsob;* [hsb] *konjunktiw;* [pl] *tryb przypuszczający;* [uk] *умовний спосіб;* [ru] *условное наклонение, кондиционал;* [sl] *pogojnik;* [hr] *mogući način, potencijal* [bg] *условно наклонение.* Table 43.

The conditional mood (both present and past) is formed periphrastically using the active (l-) participle of the content verb and a special form of the auxiliary verb *to be*. The auxiliary form is annotated `Mood=Cnd`, the participle is not. The `Tense` feature of the auxiliary is empty. Some languages have present and past conditional but the difference is expressed analytically and the same auxiliary form is used in both.

The auxiliary form is finite and in some languages (e.g. Czech) it inflects for number and person. In other languages (e.g. Russian) it has been reduced to a single frozen form that is used in all persons and numbers. Some authors may prefer to tag the frozen auxiliary as particle (`PART`), but we suggest that it be tagged `AUX`, with the verb *to be* as its lemma, to keep the annotation parallel across Slavic languages.

In Slovak and Slovenian, the reduced particle-like conditional auxiliary *by* / *bi* is used and combined with the present indicative auxiliary exactly as for the past tense (all persons in Slovenian, only 1st and 2nd in Slovak). The present auxiliary is written separately. Similar analysis can be done in Polish where the present auxiliary takes the form of the agglutinating morpheme (cf. Section 10) but is treated as an independent syntactic word: *potrafili-by-śmy* "we would be able".

Sometimes the conditional auxiliary merges with a subordinating conjunction as in Czech *aby* "so that", *kdyby* "if", Polish *żebyście* "so that you", *gdybyśmy* "if we", or Russian *чтобы (čtoby)* "so that". According to the UD guidelines we should split such fusions back into syntactic words in the annotation *(что-бы)*.

12. Adverbial Participle (Transgressive)

[cs] *přechodník přítomný, přechodník minulý;* [sk] *prechodník;* [hsb] *transgresiw;* [pl] *imiesłów przysłówkowy współczesny, imiesłów przysłówkowy uprzedni;* [uk] *дієприслівник теперішнього часу, дієприслівник минулого часу;* [ru] *деепричастие настоящего времени, деепричастие прошедшего времени;* [sl] *deležje;* [hr] *glagolski prilog sadašnji, glagolski prilog prošli;* [bg] *деепричастие.* Tables 44–52.

Adverbial participles, also called transgressives, verbal adverbs, converbs (Nedjalkov and Nedjalkov, 1987) or even gerunds (Comrie and Corbett, 2001),[10] are non-finite forms of verbs that can be used as adverbial modifiers in a clause. The circumstance they specify is that the action of the main verb happens *while* the action of the

[10] The term *gerund* may cause confusion: in English it is close to verbal nouns (cf. Section 16), in Romance languages the term denotes present participles. The term *transgressive* is unique but it is not widely known. We can encounter it in descriptions of Czech and the Sorbian languages; more generally, its usage is limited to the German-Slavic linguistic tradition. We use the term here because it is part of the UD guidelines v1, encoded as the feature `VerbForm=Trans`.

`Tense`	`Number`	`Gender`	be	can	go/come	do	accept
`Pres`	`Sing`	`Masc`	*jsa*	*moha*	*jda*	*dělaje*	*akceptuje*
`Pres`	`Sing`	`Fem,Neut`	*jsouc*	*mohouc*	*jdouc*	*dělajíc*	*akceptujíc*
`Pres`	`Plur`		*jsouce*	*mohouce*	*jdouce*	*dělajíce*	*akceptujíce*
`Past`	`Sing`	`Masc`	*byv*		*přišed*	*udělav*	*akceptovav*
`Past`	`Sing`	`Fem,Neut`	*byvši*		*přišedši*	*udělavši*	*akceptovavši*
`Past`	`Plur`		*byvše*		*přišedše*	*udělavše*	*akceptovavše*

Table 44. [cs] VERB,AUX | VerbForm=Trans. Plural forms do not distinguish gender. The present and past transgressives in the "go/come" and "do" columns are forms of different lemmas (imperfective vs. perfective).

be	can	go	do	accept
súc	*môžúc*	*idúc*	*robiac*	*akceptujúc*

Table 45. [sk] VERB,AUX | VerbForm=Trans | Tense=Pres. Modern Slovak has only the present transgressive.

`Tense`	be	can	go/come	do	accept
`Pres`		*móžo*	*dźejo*	*dźěłajo, dźěłajcy*	*akceptujo, akceptujcy*
`Past`	*bywši*		*póšowši, póšedši*	*nadźěławši*	*akceptowawši*

Table 46. [hsb] VERB,AUX | VerbForm=Trans. The present and past transgressives in the "do" column are forms of different lemmas (imperfective vs. perfective).

transgressive is happening (present transgressive), or that it happens *after* the action of the transgressive has happened (past transgressive). The subject of the clause and of the transgressive is identical.

Present transgressives tend to be created from imperfective verbs and past transgressives from perfective verbs, but exceptions exist (Academia, 1986, p. 154). Again, `Aspect` should be fixed to lemma and not used to distinguish the two transgressives. The `Tense` feature should be used instead.

Transgressives are tagged `VERB` or `AUX` but not `ADV`, and their features include `VerbForm=Trans`. In some languages they mark gender and number of the subject. In others they don't.

Tense	be	can	go/come	do	accept
Pres	*będąc*	*mogąc*	*idąc*	*robiąc*	*akceptując*
Past	*bywszy*		*poszedłszy*	*zrobiwszy*	*akceptowawszy*

Table 47. [pl] VERB,AUX | VerbForm=Trans. The present and past transgressives in the "go" and "do" columns are forms of different lemmas (imperfective vs. perfective).

Tense	be	can	go/come	do	accept
Pres	*будучи* *budučy*	*можучи* *možučy*	*йдучи* *jdučy*	*роблячи* *robljačy*	*акцептуючи* *akceptujučy*
Past	*бувши* *buvšy*	*могши* *mohšy*	*прийшовши* *pryjšovšy*	*зробивши* *zrobyvšy*	*акцептувавши* *akceptuvavšy*

Table 48. [uk] VERB,AUX | VerbForm=Trans. The present and past transgressives in the "go/come" and "do" columns are forms of different lemmas (imperfective vs. perfective).

Tense	be	can	go/come	do	accept
Pres	*будучи* *buduči*		*идя* *idja*	*делая* *delaja*	*акцептуя* *akceptuja*
Past	*быв, бывши* *byv, byvši*	*могши* *mogši*	*шедши* *šedši*	*делав, делавши* *delav, delavši*	*акцептовавши* *akceptovavši*

Table 49. [ru] VERB,AUX | VerbForm=Trans.

Tense	be	can	go/come	do	accept
Pres	*bodoč*		*idoč*	*delaje*	*akceptiraje*
Past	*bivši*		*prišedši*	*dodelavši*	*akceptiravši*

Table 50. [sl] VERB,AUX | VerbForm=Trans. The present and past transgressives in the "go/come" and "do" columns are forms of different lemmas (imperfective vs. perfective).

Tense	be	can	go/come	do	accept
Pres	*budući*	*mogući*	*idući*	*delajući*	*akceptirajući*
Past	*bivši*		*došavši*	*dodelavši*	*akceptiravši*

Table 51. [hr] VERB,AUX | VerbForm=Trans. The present and past transgressives in the "go/come" and "do" columns are forms of different lemmas (imperfective vs. perfective).

be	can	go	do	accept
бъдейки, бидейки *bădejki, bidejki*	*можейки* *možejki*	*отивайки* *otivajki*	*правейки* *pravejki*	*акцептирайки* *akceptirajki*

Table 52. [bg] VERB,AUX | VerbForm=Trans.

13. Verbal Adjective or Active Participle

[cs] *přídavné jméno slovesné činné (zpřídavnělý přechodník);* [sk] *činné príčastie;* [hsb] *prezensowy particip;* [pl] *imiesłów przymiotnikowy czynny;* [uk] *активний дієприкметник;* [ru] *действительное причастие;* [sl] *deležnik na -č, -ši;* [hr] *particip, glagolski pridjev;* [bg] *сегашно деятелно причастие.* Tables 53–61.

Active verbal adjectives (or participles) correspond to transgressives (see Section 12) and are different from the active l-participle (see Section 10). They are used attributively (not predicatively) and inflect for `Case`, except for Bulgarian that has neither long participles nor cases.

They should be tagged `ADJ`, not `VERB` or `AUX`, although their derivation from verbs is quite productive. Their lemma is the nominative singular form of the adjective, not the infinitive of the verb.

Optionally their relation to verbs may be documented using the features of `VerbForm=Part`, `Voice=Act`, `Aspect` (same as the aspect of the base verb) and `Tense` (whether they correspond to present or past transgressive). The meaning directly follows from the transgressive: [cs] *dělající* "one who is doing" (present verbal adjective); *udělavší* "one who has done" (past verbal adjective).

In standard Ukrainian, active verbal adjectives are considered ungrammatical, being a consequence of russification.[11]

[11] `http://nl.ijs.si/ME/V4/msd/html/msd.A-uk.html#msd-body.1_div.3_div.11_div.5_div.1`

Number	Sing				Plur
Gender	Masc		Neut	Fem	
Animacy	Anim	Inan			
Nom	*dělající*			*dělající*	*dělající*
Gen	*dělajícího*				*dělajících*
Dat	*dělajícímu*				*dělajícím*
Acc	*dělajícího*	*dělající*			*dělající*
Voc	*dělající*				*dělající*
Loc	*dělajícím*				*dělajících*
Ins	*dělajícím*				*dělajícími*

Table 53. [cs] `ADJ | Aspect=Imp | VerbForm=Part | Voice=Act | Tense=Pres`. The adjective dělající means “doing” and is derived from the imperfective verb dělat “to do”. The corresponding past adjective is udělavší, it is derived from the perfective verb udělat and uses the same suffixes.

Number	Sing				Plur		
Gender	Masc		Neut	Fem	Masc		Fem,Neut
Animacy	Anim	Inan			Anim	Inan	
Nom	*robiaci*		*robiace*	*robiaca*	*robiaci*	*robiace*	
Gen	*robiaceho*			*robiacej*	*robiacich*		
Dat	*robiacemu*			*robiacej*	*robiacim*		
Acc	*robiaceho*	*robiaci*	*robiace*	*robiacu*	*robiacich*	*robiace*	
Loc	*robiacom*			*robiacej*	*robiacich*		
Ins	*robiacim*			*robiacou*	*robiacimi*		

Table 54. [sk] `ADJ | Aspect=Imp | VerbForm=Part | Voice=Act | Tense=Pres`. The adjective robiaci means “doing” and is derived from the imperfective verb robiť “to do”. The corresponding past adjective is robivší with similar suffixes.

Nu	Sing				Dual			Plur		
Ge	Masc		Neut	Fem	Masc		F.,N.	Masc		F.,N.
An	An.	In.			An.	In.		An.	In.	
Nom	*dźěłacy*		*dźěłace*	*dźěłaca*	*dźěłacaj*	*dźěłacej*		*dźěłaci*	*dźěłace*	
Gen	*dźěłaceho*			*dźěłaceje*	*dźěłaceju*			*dźěłacych*		
Dat	*dźěłacemu*			*dźěłacej*	*dźěłacymaj*			*dźěłacym*		
Acc	*dźěłaceho*	*dźěłacy*	*dźěłace*	*dźěłacu*	*dźěłaceju*	*dźěłacej*		*dźěłacych*	*dźěłace*	
Loc	*dźěłacym*			*dźěłacej*	*dźěłacymaj*			*dźěłacych*		
Ins	*dźěłacym*			*dźěłacej*	*dźěłacymaj*			*dźěłacymi*		

Table 55. [hsb] `ADJ | Aspect=Imp | VerbForm=Part | Voice=Act | Tense=Pres`. The adjective dźěłacy means "doing" and is derived from the imperfective verb dźěłać "to do".

Number	Sing				Plur		
Gender	Masc		Neut	Fem	Masc		Fem,Neut
Animacy	Anim,Nhum	Inan			Anim	Nhum,Inan	
Nom	*robiący*		*robiące*	*robiąca*	*robiący*	*robiące*	
Gen	*robiącego*			*robiącej*	*robiących*		
Dat	*robiącemu*			*robiącej*	*robiącym*		
Acc	*robiącego*	*robiący*	*robiące*	*robiącą*	*robiących*	*robiące*	
Voc	*robiący*		*robiące*	*robiąca*	*robiący*	*robiące*	
Loc	*robiącym*			*robiącej*	*robiących*		
Ins	*robiącym*			*robiącą*	*robiącymi*		

Table 56. [pl] `ADJ | Aspect=Imp | VerbForm=Part | Voice=Act | Tense=Pres`. The adjective robiący means "doing" and is derived from the imperfective verb robić "to do". The corresponding past adjective is zrobiwszy, it is derived from the perfective verb zrobić and uses the same suffixes.

Number	Sing				Plur
Gender	Masc		Neut	Fem	
Animacy	Anim	Inan			
Nom	*делающий* *delajuščij*		*делающее* *delajuščee*	*делающая* *delajuščaja*	*делающие* *delajuščie*
Gen	*делающего* *delajuščego*			*делающей* *delajuščej*	*делающих* *delajuščih*
Dat	*делающему* *delajuščemu*			*делающей* *delajuščej*	*делающим* *delajuščim*
Acc	*делающего* *delajuščego*	*делающий* *delajuščij*	*делающее* *delajuščee*	*делающую* *delajuščuju*	*делающие* *delajuščie*
Loc	*делающем* *delajuščem*			*делающей* *delajuščej*	*делающих* *delajuščih*
Ins	*делающим* *delajuščim*			*делающей, делающею* *delajuščej, delajuščeju*	*делающими* *delajuščimi*

Table 57. [ru] ADJ | Aspect=Imp | VerbForm=Part | Voice=Act | Tense=Pres. The adjective делающий (delajuščij) means "doing" and is derived from the imperfective verb делать (delat') "to do". The corresponding past adjective is сделавший (sdelavšij), it is derived from the perfective verb сделать (sdelat') and uses the same suffixes.

Nu	Sing				Dual		Plur		
Ge	Masc		Neut	Fem	Masc	Fem,Neut	Masc	Fem	Neut
An	Anim	Inan							
Nom	*delajoč*		*delajoče*	*delajoča*	*delajoča*	*delajoči*	*delajoči*	*delajoče*	*delajoča*
Gen	*delajočega*			*delajoče*	*delajočih*				
Dat	*delajočemu*			*delajoči*	*delajočima*		*delajočim*		
Acc	*delajočega*	*delajoč*	*delajoče*	*delajočo*	*delajoča*	*delajoči*	*delajoče*		*delajoča*
Loc	*delajočem*			*delajoči*	*delajočih*				
Ins	*delajočim*			*delajočo*	*delajočima*		*delajočimi*		

Table 58. [sl] ADJ | Aspect=Imp | VerbForm=Part | Voice=Act | Tense=Pres. The adjective delajoč / delajoči means "doing" and is derived from the imperfective verb delati "to do".

Number	Sing				Plur		
Gender	Masc		Neut	Fem	Masc	Fem	Neut
Animacy	Anim	Inan					
Nom	*delajući*		*delajuće*	*delajuća*	*delajući*	*delajuće*	*delajuća*
Gen	*delajućeg*			*delajuće*	*delajućih*		
Dat	*delajućem*			*delajućoj*	*delajućim*		
Acc	*delajućeg*	*delajući*	*delajuće*	*delajuću*	*delajuće*		*delajuća*
Voc	*delajući*		*delajuće*	*delajuća*	*delajući*	*delajuće*	*delajuća*
Loc	*delajućem*			*delajućoj*	*delajućim*		
Ins	*delajućim*			*delajućom*	*delajućim*		

Table 59. [hr] ADJ | Aspect=Imp | VerbForm=Part | Voice=Act | Tense=Pres. The adjective delajući means "doing" and is derived from the imperfective verb delati "to do". The corresponding past adjective is dodelavši, it is derived from the perfective verb dodelati and uses the same suffixes.

Number	Sing			Plur
Gender	Masc	Fem	Neut	
Ind	*правещ* *pravešt*	*правеща* *pravešta*	*правещо* *pravešto*	*правещи* *pravešti*
Def	*правещият* *praveštijat*	*правещата* *praveštata*	*правещото* *praveštoto*	*правещите* *praveštite*

Table 60. [bg] правещ (pravešt) "doing" ADJ | Aspect=Imp | VerbForm=Part | Voice=Act | Tense=Pres. The rows correspond to different values of Definite. Bulgarian adjectives do not inflect for Case.

<table>
<tr><td>Number</td><td colspan="3">Sing</td><td colspan="3">Dual</td></tr>
<tr><td>Gender</td><td>Masc</td><td>Neut</td><td>Fem</td><td>Masc</td><td>Neut</td><td>Fem</td></tr>
<tr><td>Nom</td><td colspan="2">дѣлаѩ
dělaję</td><td>дѣлаѭщи
dělajǫšti</td><td>дѣлаѭща
dělajǫšta</td><td colspan="2">дѣлаѭщи
dělajǫšti</td></tr>
<tr><td>Gen</td><td colspan="2">дѣлаѭща
dělajǫšta</td><td>дѣлаѭщѧ
dělajǫštę</td><td colspan="3">дѣлаѭщоу
dělajǫštu</td></tr>
<tr><td>Dat</td><td colspan="2">дѣлаѭщоу
dělajǫštu</td><td>дѣлаѭщи
dělajǫšti</td><td colspan="2">дѣлаѭщема
dělajǫštema</td><td>дѣлаѭщама
dělajǫštama</td></tr>
<tr><td>Acc</td><td>дѣлаѭщь
dělajǫštь</td><td>дѣлаѭще
dělajǫšte</td><td>дѣлаѭщѫ
dělajǫštǫ</td><td>дѣлаѭща
dělajǫšta</td><td colspan="2">дѣлаѭщи
dělajǫšti</td></tr>
<tr><td>Voc</td><td colspan="2">дѣлаѩ
dělaję</td><td>дѣлаѭщи
dělajǫšti</td><td>дѣлаѭща
dělajǫšta</td><td colspan="2">дѣлаѭщи
dělajǫšti</td></tr>
<tr><td>Loc</td><td colspan="3">дѣлаѭщи
dělajǫšti</td><td colspan="3">дѣлаѭщоу
dělajǫštu</td></tr>
<tr><td>Ins</td><td colspan="2">дѣлаѭщемь
dělajǫštemь</td><td>дѣлаѭщеѭ
dělajǫštejǫ</td><td colspan="2">дѣлаѭщема
dělajǫštema</td><td>дѣлаѭщама
dělajǫštama</td></tr>
</table>

…

…

<table>
<tr><td>Number</td><td colspan="3">Plur</td></tr>
<tr><td>Gender</td><td>Masc</td><td>Neut</td><td>Fem</td></tr>
<tr><td>Nom</td><td>дѣлаѭще
dělajǫšte</td><td>дѣлаѭща
dělajǫšta</td><td>дѣлаѭщѧ
dělajǫštę</td></tr>
<tr><td>Gen</td><td colspan="3">дѣлаѭщь
dělajǫštь</td></tr>
<tr><td>Dat</td><td colspan="2">дѣлаѭщемъ
dělajǫštemъ</td><td>дѣлаѭщамъ
dělajǫštamъ</td></tr>
<tr><td>Acc</td><td>дѣлаѭщѧ
dělajǫštę</td><td>дѣлаѭща
dělajǫšta</td><td>дѣлаѭщѧ
dělajǫštę</td></tr>
<tr><td>Voc</td><td>дѣлаѭще
dělajǫšte</td><td>дѣлаѭща
dělajǫšta</td><td>дѣлаѭщѧ
dělajǫštę</td></tr>
<tr><td>Loc</td><td colspan="2">дѣлаѭщихъ
dělajǫštichъ</td><td>дѣлаѭщахъ
dělajǫštachъ</td></tr>
<tr><td>Ins</td><td colspan="2">дѣлаѭщи
dělajǫšti</td><td>дѣлаѭщами
dělajǫštami</td></tr>
</table>

Table 61. [cu] ADJ | Aspect=Imp | VerbForm=Part | Voice=Act | Tense=Pres. The adjective дѣлаѩ *(dělaję) means “doing” and is derived from the imperfective verb* дѣлати *(dělati) “to do”. The corresponding past adjective is* съдѣлавъ *(sъdělavъ), it is derived from the perfective verb* съдѣлати *(sъdělati) and uses similar suffixes: Sing Masc Gen* съдѣлавъша *(sъdělavъša), Sing Fem Nom* съдѣлавъши *(sъdělavъši) etc. The table shows the short (“strong”) forms of the nominal declension.*

14. Passive Participle

[cs] *příčestí trpné, přídavné jméno slovesné trpné;* [sk] *trpné príčastie;* [hsb] *preteritowy particip;* [pl] *imiesłów przymiotnikowy bierny;* [uk] *пасивний дієприкметник;* [ru] *страдательное причастие;* [sl] *trpni deležnik;* [hr] *glagolski pridjev trpni;* [bg] *минало страдателно причастие.* Tables 62–72.

The passive participle is a non-finite verbal form used to construct the periphrastic passive. It is the only form that bears the feature `Voice=Pass`.

All the other verb forms may take part in passive constructions. Examples [cs]: *je nominován* "he is (being) nominated"; *byl jsem nominován* "I was nominated"; *byl bych nominován* "I would be nominated"; *budeš nominován* "you will be nominated"; *buďte nominován* "be nominated"; *být nominován* "to be nominated" etc. It is always the passive participle that makes the construction passive. The auxiliary verb forms do not differ morphologically from the forms used in the active voice, which is the default. Therefore they should either be marked `Voice=Act`, or the `Voice` feature should be left empty. We suggest that the explicit annotation of `Voice=Act` is mandatory for the other participles, so that all types of participles are explicitly distinguished. For the other verbal forms, the feature is optional.

Note that Slavic languages also have the reflexive passive, consisting of a reflexive pronoun and a 3rd person indicative verb ([cs] *Prezident* ***se volí*** *každé 4 roky.* "The president is elected every 4 years.") Although the analytical construction is passive, the participating verb is morphologically not passive and will not be marked as such. The passive nature of the clause will be visible in the dependency annotation (the subject will be attached as `nsubjpass` and the reflexive pronoun will be attached using the language-specific relation `auxpass:reflex`). In [ru] the reflexive pronoun is written as one word with the finite verb: *негласно считало**сь**, что ему простительно всякое (neglasno sčitalo**s'**, čto emu prostitel'no vsjakoe)* "it was silently thought that he could be forgiven everything". When it is used to form the reflexive passive, we could in theory mark the whole form as passive; however, we recommend to split the form to two syntactic words *(считало+сь / sčitalo+s')* and make it parallel with the other Slavic languages.

Passive participles may have short and long forms. As explained above (see Section 3), this distinction can be interpreted as indefinite vs. definite adjectives in the south Slavic languages. In the north it applies to Czech and Russian, where the short forms are used predicatively, and their `Case` inflection almost vanished (Czech short participles may form accusative but it is very rare). Since we cannot distinguish the forms by the `Definite` feature here, we suggest to tag the short forms `VERB`, even though the remnants of case inflection make this decision slightly inconsistent with the rest.[12] The long forms are also called passive verbal adjectives and we treat them

[12]We also lose the parallelism between short passive participles and short forms of adjectives in Czech (*nemocen* vs. *nemocný* "ill"). The short adjectives are used in predicates as well. This is a controversial issue and the guideline we propose may be revised in future.

Number	Sing				Plur			
Gender	Masc		Neut	Fem	Masc		Fem	Neut
Animacy	Anim	Inan			Anim	Inan		
Nom	*dělaný*		*dělané*	*dělaná*	*dělaní*	*dělané*		*dělaná*
Gen	*dělaného*			*dělané*	*dělaných*			
Dat	*dělanému*			*dělané*	*dělaným*			
Acc	*dělaného*	*dělaný*	*dělané*	*dělanou*	*dělané*			*dělaná*
Voc	*dělaný*		*dělané*	*dělaná*	*dělaní*	*dělané*		*dělaná*
Loc	*dělaném*			*dělané*	*dělaných*			
Ins	*dělaným*			*dělanou*	*dělanými*			
VERB	*dělán*		*děláno*	*dělána*	*děláni*	*dělány*		*dělána*

Table 62. [cs] dělaný / dělán "done" `ADJ,VERB | Aspect=Imp | VerbForm=Part | Voice=Pass`.

Number	Sing				Plur		
Gender	Masc		Neut	Fem	Masc		Fem,Neut
Animacy	Anim	Inan			Anim	Inan	
Nom	*robený*		*robené*	*robená*	*robení*	*robené*	
Gen	*robeného*			*robenej*	*robených*		
Dat	*robenému*			*robenej*	*robeným*		
Acc	*robeného*	*robený*	*robené*	*robenú*	*robených*	*robené*	
Loc	*robenom*			*robenej*	*robených*		
Ins	*robeným*			*robenou*	*robenými*		

Table 63. [sk] robený "done" `ADJ | Aspect=Imp | VerbForm=Part | Voice=Pass`.

as adjectives derived from verbs. Their tag should be `ADJ` and their lemma should be the adjectival form in masculine singular nominative, not the verb infinitive. They can be used as attributive modifiers of noun phrases (with which they agree in gender, number and case).

The long forms of passive participles may also be used in predicates, especially in languages that have only the long forms (e.g. Slovak). However, since they are tagged as adjectives, the dependency layer will analyze them as adjectival predicates with a copula.

In Polish and Ukrainian, the attributive form of singular neuter is different from the predicative one: [uk] *писа**не** правило (pysa**ne** pravylo)* "a written rule" vs. *правило*

<table>
<tr><td>Nu</td><td colspan="4">Sing</td><td colspan="3">Dual</td><td colspan="3">Plur</td></tr>
<tr><td>Ge</td><td colspan="2">Masc</td><td>Neut</td><td>Fem</td><td colspan="2">Masc</td><td>F.,N.</td><td colspan="2">Masc</td><td>F.,N.</td></tr>
<tr><td>An</td><td>An.</td><td>In.</td><td></td><td></td><td>An.</td><td>In.</td><td></td><td>An.</td><td>In.</td><td></td></tr>
<tr><td>Nom</td><td colspan="2">dźěłany</td><td>dźěłane</td><td>dźěłana</td><td>dźěłanaj</td><td colspan="2">dźěłanej</td><td>dźěłani</td><td colspan="2">dźěłane</td></tr>
<tr><td>Gen</td><td colspan="3">dźěłaneho</td><td>dźěłaneje</td><td colspan="3">dźěłaneju</td><td colspan="3">dźěłanych</td></tr>
<tr><td>Dat</td><td colspan="3">dźěłanemu</td><td>dźěłanej</td><td colspan="3">dźěłanymaj</td><td colspan="3">dźěłanym</td></tr>
<tr><td>Acc</td><td>dźěłaneho</td><td>dźěłany</td><td>dźěłane</td><td>dźěłanu</td><td>dźěłaneju</td><td colspan="2">dźěłanej</td><td>dźěłanych</td><td colspan="2">dźěłane</td></tr>
<tr><td>Loc</td><td colspan="3">dźěłanym</td><td>dźěłanej</td><td colspan="3">dźěłanymaj</td><td colspan="3">dźěłanych</td></tr>
<tr><td>Ins</td><td colspan="3">dźěłanym</td><td>dźěłanej</td><td colspan="3">dźěłanymaj</td><td colspan="3">dźěłanymi</td></tr>
</table>

Table 64. [hsb] dźěłany "done" ADJ | Aspect=Imp | VerbForm=Part | Voice=Pass.

<table>
<tr><td>Number</td><td colspan="4">Sing</td><td colspan="3">Plur</td></tr>
<tr><td>Gender</td><td colspan="2">Masc</td><td>Neut</td><td>Fem</td><td colspan="2">Masc</td><td>Fem,Neut</td></tr>
<tr><td>Animacy</td><td>Anim,Nhum</td><td>Inan</td><td></td><td></td><td>Anim</td><td>Nhum,Inan</td><td></td></tr>
<tr><td>Nom</td><td colspan="2">robiony</td><td>robione</td><td>robiona</td><td>robieni</td><td colspan="2">robione</td></tr>
<tr><td></td><td colspan="2"></td><td>robiono</td><td></td><td colspan="3"></td></tr>
<tr><td>Gen</td><td colspan="3">robionego</td><td>robionej</td><td colspan="3">robionych</td></tr>
<tr><td>Dat</td><td colspan="3">robionemu</td><td>robionej</td><td colspan="3">robionym</td></tr>
<tr><td>Acc</td><td>robionego</td><td>robiony</td><td>robione</td><td>robioną</td><td>robionych</td><td colspan="2">robione</td></tr>
<tr><td>Voc</td><td colspan="2">robiony</td><td>robione</td><td>robiona</td><td>robieni</td><td colspan="2">robione</td></tr>
<tr><td>Loc</td><td colspan="3">robionym</td><td>robionej</td><td colspan="3">robionych</td></tr>
<tr><td>Ins</td><td colspan="3">robionym</td><td>robioną</td><td colspan="3">robionymi</td></tr>
</table>

Table 65. [pl] robiony "done" ADJ | Aspect=Imp | VerbForm=Part | Voice=Pass.

*писа**но** (pravylo pysa**no**)* "a rule is/was written". One might be tempted to tag the predicative forms as `VERB` instead of `ADJ`, to make them parallel with the short (predicative) participles in Czech and Russian. Unfortunately, that would mean that two very similar Ukrainian sentences would get different part-of-speech and dependency analyses just because their subjects differ in gender and/or number. Therefore it seems better to classify these forms as adjectives, too.

Slovenian and Serbo-Croatian inflect both short and long adjectives for `Case`, and the same applies to passive participles (passive verbal adjectives).

Definite adjectives are longer than indefinite also in Bulgarian and Macedonian, although the construction is different from that of [sl] and [hr]. The definite forms are used only attributively, the short forms both as attributes and predicates. As this

Number	Sing				Plur
Ge/An	M/Anim	M/Inan	Neut	Fem	
Nom	*делаемый* *delaemyj*		*делаемое* *delaemoe*	*делаемая* *delaemaja*	*делаемые* *delaemye*
Gen	*делаемого* *delaemogo*			*делаемой* *delaemoj*	*делаемых* *delaemyh*
Dat	*делаемому* *delaemomu*			*делаемой* *delaemoj*	*делаемым* *delaemym*
Acc	*делаемого* *delaemogo*	*делаемый* *delaemyj*	*делаемое* *delaemoe*	*делаемую* *delaemuju*	*делаемых, делаемые* *delaemyh, delaemye*
Loc	*делаемом* *delaemom*			*делаемой* *delaemoj*	*делаемых* *delaemyh*
Ins	*делаемым* *delaemym*			*делаемой, делаемою* *delaemoj, delaemoju*	*делаемыми* *delaemymi*
VERB	*делаем* *delaem*		*делаемо* *delaemo*	*делаема* *delaema*	*делаемы* *delaemy*

Table 66. [ru] ADJ,VERB | Aspect=Imp | VerbForm=Part | Voice=Pass | Tense=Pres.

Number	Sing				Plur
Ge/An	M/Anim	M/Inan	Neut	Fem	
Nom	*сделанный* *sdelannyj*		*сделанное* *sdelannoe*	*сделанная* *sdelannaja*	*сделанные* *sdelannye*
Gen	*сделанного* *sdelannogo*			*сделанной* *sdelannoj*	*сделанных* *sdelannyh*
Dat	*сделанному* *sdelannomu*			*сделанной* *sdelannoj*	*сделанным* *sdelannym*
Acc	*сделанного* *sdelannogo*	*сделанный* *sdelannyj*	*сделанное* *sdelannoe*	*сделанную* *sdelannuju*	*сделанных, сделанные* *sdelannyh, sdelannye*
Loc	*сделанном* *sdelannom*			*сделанной* *sdelannoj*	*сделанных* *sdelannyh*
Ins	*сделанным* *sdelannym*			*сделанной, сделанною* *sdelannoj, sdelannoju*	*сделанными* *sdelannymi*
VERB	*сделан* *sdelan*		*сделано* *sdelano*	*сделана* *sdelana*	*сделаны* *sdelany*

Table 67. [ru] сделанный / сделан (sdelannyj / sdelan) "done" ADJ,VERB | Aspect=Perf | VerbForm=Part | Voice=Pass | Tense=Past.

Number	Sing				Plur	
Gender	Masc		Neut	Fem		
Animacy	Anim	Inan			Anim	Inan
Nom	*зроблений* *zroblenyj*		*зроблене* *zroblene*	*зроблена* *zroblena*	*зроблені* *zrobleni*	
			зроблено *zrobleno*			
Gen	*зробленого* *zroblenoho*			*зробленої* *zroblenoï*	*зроблених* *zroblenych*	
Dat	*зробленому* *zroblenomu*			*зробленій* *zroblenij*	*зробленим* *zroblenym*	
Acc	*зробленого* *zroblenoho*	*зроблений* *zroblenyj*	*зроблене* *zroblene*	*зроблену* *zroblenu*	*зроблених* *zroblenych*	*зроблені* *zrobleni*
Loc	*зробленому* *zroblenomu*			*зробленій* *zroblenij*	*зроблених* *zroblenych*	
Ins	*зробленим* *zroblenym*			*зробленою* *zroblenoju*	*зробленими* *zroblenymy*	

Table 68. [uk] зроблений (zroblenyj) "done" `ADJ | Aspect=Perf | VerbForm=Part | Voice=Pass`. The `Nom`-`Ins` rows show `Case` inflections of verbal adjectives.

Number	Sing			Plur
Gender	Masc	Fem	Neut	
Ind	*правен* *praven*	*правена* *pravena*	*правено* *praveno*	*правени* *praveni*
Def	*правеният* *pravenijat*	*правената* *pravenata*	*правеното* *pravenoto*	*правените* *pravenite*

Table 69. [bg] правен (praven) "done" `ADJ | Aspect=Imp | VerbForm=Part | Voice=Pass`. The rows correspond to different values of `Definite`. Bulgarian adjectives do not inflect for `Case`.

also applies to passive participles, it seems appropriate to classify them (both forms) as `ADJ`. They do not inflect for `Case` but neither do adjectives because [bg] and [mk] have lost the case system.

Russian and Old Church Slavonic distinguish present and past passive participles: *журнал, читаемый студентом (žurnal, čitaemyj studentom)* "journal that is being read by the student" vs. *журнал, прочитанный студентом (žurnal, pročitannyj studentom)*

Nu	Sing				Dual		Plur		
Ge	Masc		Neut	Fem	Masc	Fem,Neut	Masc	Fem	Neut
An	Anim	Inan							
Nom	*delan*		*delano*	*delana*	*delana*	*delani*	*delani*	*delane*	*delana*
Gen	*delanega*			*delane*	*delanih*				
Dat	*delanemu*			*delani*	*delanima*		*delanim*		
Acc	*delanega*	*delan*	*delano*		*delana*	*delani*	*delane*		*delana*
Loc	*delanem*			*delani*	*delanih*				
Ins	*delanim*			*delano*	*delanima*		*delanimi*		

Table 70. [sl] delan / delani “done” ADJ | Aspect=Imp | VerbForm=Part | Voice=Pass.

Number	Sing				Plur		
Gender	Masc		Neut	Fem	Masc	Fem	Neut
Animacy	Anim	Inan					
Nom	*delan*		*delano*	*delana*	*delani*	*delane*	*delana*
Gen	*delanog*			*delane*	*delanih*		
Dat	*delanom*			*delanoj*	*delanim*		
Acc	*delanog*	*delan*	*delano*	*delanu*	*delane*		*delana*
Voc	*delan*		*delano*	*delana*	*delani*	*delane*	*delana*
Loc	*delanom*			*delanoj*	*delanim*		
Ins	*delanim*			*delanom*	*delanim*		

Table 71. [hr] delan / delani “done” ADJ | Aspect=Imp | VerbForm=Part | Voice=Pass.

“journal that has been read by the student”. The distinction will be annotated using the Tense feature. Note that other languages will have the Tense feature empty. Both the above examples will use the same (the only) passive participle in Czech, they will differ only by the prefix because the second verb is perfective: *časopis (pře)čtený studentem* “journal read by the student”.

Passive participles are normally formed for transitive verbs, although verbs that subcategorize for a non-accusative object may also have a passive participle (neuter singular only).

Number	Sing			Dual		
Gender	Masc	Neut	Fem	Masc	Neut	Fem
Nom	дѣлаемъ *dělajem*	дѣлаемо *dělajemo*	дѣлаема *dělajema*	дѣлаема *dělajema*	дѣлаемѣ *dělajemě*	
Gen	дѣлаема *dělajema*		дѣлаемы *dělajemy*	дѣлаемоу *dělajemu*		
Dat	дѣлаемоу *dělajemu*		дѣлаемѣ *dělajemě*	дѣлаемома *dělajemoma*		дѣлаемама *dělajemama*
Acc	дѣлаемъ *dělajemъ*	дѣлаемо *dělajemo*	дѣлаемѫ *dělajemǫ*	дѣлаема *dělajema*	дѣлаемѣ *dělajemě*	
Voc	дѣлаемъ *dělajemъ*	дѣлаемо *dělajemo*		дѣлаема *dělajema*	дѣлаемѣ *dělajemě*	
Loc	дѣлаемѣ *dělajemě*			дѣлаемоу *dělajemu*		
Ins	дѣлаемомь *dělajemomь*		дѣлаемоѭ *dělajemojǫ*	дѣлаемома *dělajemoma*		дѣлаемама *dělajemama*

…

…

Number	Plur		
Gender	Masc	Neut	Fem
Nom	дѣлаеми *dělajemi*	дѣлаема *dělajema*	дѣлаемы *dělajemy*
Gen	дѣлаемъ *dělajemъ*		
Dat	дѣлаемомъ *dělajemomъ*		дѣлаемамъ *dělajemamъ*
Acc	дѣлаемы *dělajemy*	дѣлаема *dělajema*	дѣлаемы *dělajemy*
Voc	дѣлаеми *dělajemi*	дѣлаема *dělajema*	дѣлаемы *dělajemy*
Loc	дѣлаемѣхъ *dělajǫštichъ*		дѣлаемахъ *dělajǫštachъ*
Ins	дѣлаемы *dělajemy*		дѣлаемами *dělajemami*

Table 72. [cu] ADJ | Aspect=Imp | VerbForm=Part | Voice=Pass | Tense=Pres. The adjective дѣлаем *(dělajem) means "being done" and is derived from the imperfective verb* дѣлати *(dělati) "to do". The corresponding past adjective is* съдѣлан *(sъdělan) "done", it is derived from the perfective verb* съдѣлати *(sъdělati) and uses similar suffixes: Sing Fem Nom* съдѣлана *(sъdělana), Sing Neut Nom* съдѣлано *(sъdělano) etc. The table shows the short ("strong") forms of the nominal declension.*

Example	Gloss	Languages	L	Tag	VerbFo	Voic	Tense	Defin
budoucí	*what will be*	all?	l	ADJ	(Part)		(Fut)	
delajoč	*who is doing*	sl, bg, cu	s	ADJ	Part	Act	Pres	Ind
dělající	*who is doing*	all	l	ADJ	Part	Act	Pres	(Def)
съдѣлавъ	*who has done*	cu	s	ADJ	Part	Act	Past	Ind
udělavší	*who has done*	cs, sk, hsb, pl uk, ru, hr	l	ADJ	Part	Act	Past	(Def)
dělal	*did / (has) done*	all	s	VERB AUX	Part	Act	Past	
правел	*was doing*	bg	s	VERB	Part	Act	Imp	
minulý	*what has passed*	all?	l	ADJ	(Part)		(Past)	
dělán	*(is (being)) done*	cs	s	VERB	Part	Pass		
delan	*((who) is) done*	sl, hr, bg	s	ADJ	Part	Pass		Ind
dělaný	*who is/was done*	cs, sk, hsb, pl, uk, sl, hr, bg	l	ADJ	Part	Pass		(Def)
делаем	*(is being) done*	ru	s	VERB	Part	Pass	Pres	
дѣлаемъ	*(is being) done*	cu	s	ADJ	Part	Pass	Pres	Ind
делаемый	*who is being done*	ru, cu	l	ADJ	Part	Pass	Pres	(Def)
сделан	*(has been, is) done*	ru	s	VERB	Part	Pass	Past	
съдѣлан	*(who is) done*	cu	s	ADJ	Part	Pass	Past	Ind
сделанный	*who has been done*	ru, cu	l	ADJ	Part	Pass	Past	(Def)

Table 73. Participles. The "L" column denotes short vs. long forms. The Def feature only applies in languages where the Ind counterpart exists.

15. Participle Summary

Participles are words that share properties of verbs and adjectives. Just like adjectives, they have short and long forms. Historically, the long forms emerged as a fusion of the short form and a pronoun. North Slavic languages either do not have the short form or they do not mark the Case on it. Short and long forms are distinguished by the POS tag (VERB/ADJ). South Slavic languages use the short form and inflect it for Case (except for [bg] and [mk], which have lost cases). The long form is definite. Both forms are ADJ; short vs. long is distinguished by Definite=Ind/Def. The l-participle is special. Its short form is VERB even in the south Slavic languages (the Definite and Case features of the short form are empty). Table 73 gives a summary of the proposed annotation of participles. Adverbial participles are not covered here because we tag them as transgressives (VerbForm=Trans, see Section 12). [cu] does not have transgres-

Number	Sing	Plur
Nom	*dělání*	*dělání*
Gen	*dělání*	*dělání*
Dat	*dělání*	*děláním*
Acc	*dělání*	*dělání*
Voc	*dělání*	*dělání*
Loc	*dělání*	*děláních*
Ins	*děláním*	*děláními*

Table 74. [cs] dělání "doing" NOUN | Aspect=Imp. The rows correspond to different values of Case.

Number	Sing	Plur
Nom	*robenie*	*robenia*
Gen	*robenia*	*robení*
Dat	*robeniu*	*robeniam*
Acc	*robenie*	*robenia*
Loc	*robení*	*robeniach*
Ins	*robením*	*robeniami*

Table 75. [sk] robenie "doing" NOUN | Aspect=Imp. The rows correspond to different values of Case.

sives but the nominative forms of its active participles correspond to transgressives and can be used as adverbial modifiers.

16. Verbal Noun

[cs] *podstatné jméno slovesné;* [sk] *slovesné podstatné meno;* [hsb] *werbalny substantiw;* [pl] *rzeczownik odczasownikowy;* [uk] *віддієслівний іменник;* [ru] *отглагольное существительное;* [sl] *glagolsko ime;* [hr] *radna (glagolska) imenica;* [bg] *отглаголно съществително име.* Tables 74–83.

Verbal noun is an abstract noun productively derived from a verb, denoting the action of the verb. It inflects for Case and Number, although it is only rarely seen in plural. Its gender is always Neut. We tag it NOUN and use its singular nominative form as the lemma (not the infinitive of the base verb).

The UD guidelines v1 suggest that VerbForm=Ger can be used to distinguish verbal nouns from other nouns. This works in English where the corresponding form is

Number	Sing	Dual	Plur
Nom	*dźěłanje*	*dźěłani*	*dźěłanja*
Gen	*dźěłanja*	*dźěłanjow*	
Dat	*dźěłanju*	*dźěłanjomaj*	*dźěłanjam*
Acc	*dźěłanje*	*dźěłani*	*dźěłanja*
Loc	*dźěłanju*	*dźěłanjomaj*	*dźěłanjach*
Ins	*dźěłanjom*	*dźěłanjomaj*	*dźěłanjemi*

Table 76. [hsb] dźěłanje "doing" `NOUN | Aspect=Imp`. The rows correspond to different values of `Case`.

Number	Sing	Plur
Nom	*robienie*	*robienia*
Gen	*robienia*	*robień*
Dat	*robieniu*	*robieniom*
Acc	*robienie*	*robienia*
Voc	*robienie*	*robienia*
Loc	*robieniu*	*robieniach*
Ins	*robieniem*	*robieniami*

Table 77. [pl] robienie "doing" `NOUN | Aspect=Imp`. The rows correspond to different values of `Case`.

termed *gerund*. Unfortunately, this feature might cause confusion in Slavic linguistics where some authors use the term *gerund* for adverbial participles (cf. Section 12). Hence we advise against using it with Slavic verbal nouns. Nevertheless, the verbal nouns may mark the `Aspect` of their base verb.

Verbal nouns use suffixes similar to passive participles. Unlike passive participles, they can be derived from intransitive verbs as well.

17. Negation

Slavic verbs are negated by a local variant of the morpheme *ne*, which is either a bound morpheme (prefix), or a separate word (particle). If it is a prefix, we do not cut it off during tokenization.

A standalone negating word is tagged `PART` and it has the feature `Negative=Neg`. On the dependency level, it is attached to the negated verb using the `neg` relation.

Number	Sing	Plur
Nom	*роблення* *roblennja*	*роблення* *roblennja*
Gen	*роблення* *roblennja*	*роблень* *roblen'*
Dat	*робленню* *roblennju*	*робленням* *roblennjam*
Acc	*роблення* *roblennja*	*роблення* *roblennja*
Loc	*робленні, робленню* *roblenni, roblennju*	*робленнях* *roblennjach*
Ins	*робленням* *roblennjam*	*робленнями* *roblennjamy*

Table 78. [uk] роблення (roblennja) "doing" `NOUN | Aspect=Imp`. The rows correspond to different values of `Case`.

In the case of the negative prefix, the verb itself bears the `Negative=Neg` feature. This type of prefixing is considered inflectional rather than derivational, that is, the lemma is still the affirmative (unprefixed) infinitive. If the language negates verbs by prefixing, all affirmative forms of these verbs should be annotated `Negative=Pos`.

In periphrastic constructions it is normal that only one participating word is negated, but various languages may have different rules on what participant it should be. Cf. [cs] *Včera jsem **ne**šel domů.* "I did not go home yesterday." (negated participle) and [hr] *Jučer **ni**sam išao kući.* (negated auxiliary).

Verbal adjectives (long forms of participles) and verbal nouns are negated in a similar fashion.

Czech is an example of a language where all verbs are negated using the prefix *ne-*. Russian is an example of the opposite: all finite forms and the l-participles are negated using the particle *не (ne)*. With the other participles it becomes a prefix though: *несовершенный (nesoveršennyj)* "imperfect". Yet different is Croatian where the negative particle is the default, except for the verbs *biti*, *htjeti* and *imati* that take the negative morpheme as a prefix.

18. Current Data

UD version 1.2, released in November 2015, contains data from 6 Slavic languages: Czech, Polish, Slovenian, Croatian, Bulgarian and Old Church Slavonic. Most of these datasets distinguish `AUX` from `VERB` (except for [cu], which uses only the `VERB` tag) and most of them have a non-empty value of `VerbForm` for all verbs (auxiliary or not). Here

Number	Sing	Plur
Nom	*делание, деланье* *delanie, delan'e*	*делания, деланья* *delanija, delan'ja*
Gen	*делания, деланья* *delanija, delan'ja*	*деланий* *delanij*
Dat	*деланию, деланью* *delaniju, delan'ju*	*деланиям, деланьям* *delanijam, delan'jam*
Acc	*делание, деланье* *delanie, delan'e*	*делания, деланья* *delanija, delan'ja*
Loc	*делании, деланье, деланьи* *delanii, delan'e, delan'i*	*деланиях, деланьях* *delanijah, delan'jah*
Ins	*деланием, деланьем* *delaniem, delan'em*	*деланиями, деланьями* *delanijami, delan'jami*

Table 79. [ru] делание (delanie) "doing" NOUN | Aspect=Imp. The rows correspond to different values of Case.

the exceptions are [hr] (finite verbs are not marked), [pl] (predicative nonverbs such as *to* "it (is)" are tagged VERB) and [bg] (empty VerbForms are probably annotation errors). [cu] uses the subjunctive mood (Mood=Sub) instead of Mood=Cnd for the conditional auxiliaries.

All but [bg] have occurrences of VerbForm=Inf, [cu] and [sl] also have VerbForm=Sup.

All languages except [pl] tag verbal nouns as regular NOUN, without setting the VerbForm. Polish tags them VERB with VerbForm=Ger.

VerbForm=Trans is used in [cs], [pl] and [sl]; In Czech and Polish their main part of speech is VERB (or AUX) while in Slovenian it is ADV. Croatian data ignores the Trans value and annotates transgressives as ADV plus VerbForm=Part. Bulgarian tags them as regular adverbs, without any distinctive feature.

By far the largest proportion of inconsistency is caused by participles.

[cs]: The l-participles are tagged VERB/AUX VerbForm=Part | Tense=Past | Voice=Act. Short forms of passive participles are tagged VERB VerbForm=Part | Voice=Pass (empty Tense). Long forms are tagged as regular adjectives (empty VerbForm). Active participles related to transgressives are tagged ADJ VerbForm=Part | Voice=Act and distinguished by tense and aspect: either Aspect=Imp | Tense=Pres or Aspect=Perf | Tense=Past.

[pl]: All participles are tagged VERB. Present active (progressive) participles are marked Voice=Act | Tense=Pres, while the passive participles have Voice=Pass and empty Tense. The l-participles are marked as finite forms (VerbForm=Fin instead of Part!) with Tense=Past and empty Voice.

Number	Sing	Dual	Plur
Nom	*delanje*	*delanji*	*delanja*
Gen	*delanja*	*delanj*	
Dat	*delanju*	*delanjema*	*delanjem*
Acc	*delanje*	*delanji*	*delanja*
Loc	*delanju*	*delanjih*	
Ins	*delanjem*	*delanjema*	*delanji*

Table 80. [sl] delanje “doing” `NOUN | Aspect=Imp`. The rows correspond to different values of `Case`.

Number	Sing	Plur
Nom	*delanje*	*delanja*
Gen	*delanja*	*delanja*
Dat	*delanju*	*delanjima*
Acc	*delanje*	*delanja*
Voc	*delanje*	*delanja*
Loc	*delanju*	*delanjima*
Ins	*delanjem*	*delanjima*

Table 81. [hr] delanje “doing” `NOUN | Aspect=Imp`. The rows correspond to different values of `Case`.

[sl]: The predicatively used l-participles are tagged `VERB/AUX VerbForm=Part`, with empty `Voice` and `Tense`. Participles tagged as adjectives (`ADJ VerbForm=Part`) are mostly passive participles, albeit their `Voice` feature is empty, too. However, some of them are adjectives derived from the l-participles *(minuli, ostali, odrasle)* and rarely also the present active participle *(boleče)*.

[hr]: The l-participles are tagged `VERB/AUX VerbForm=Part` and they are the only active participles marked. Passive participles are tagged `ADJ VerbForm=Part`. The `Tense` and `Voice` features are always empty.

[bg]: Only the l-participles of the verb *to be* are tagged `VERB/AUX VerbForm=Part`. Predicatively used l-participles of other verbs appear as finite verbs (`VerbForm=Fin`), they are thus indistinguishable from the aorist and imperfect simple past tenses, respectively. For example, both *можах* and *могъл* (aorist and perfect l-participle of *could*) are annotated `Voice=Act | Tense=Past`. In parallel, both *можех* and *можел* (simple imperfect and imperfect l-participle of the same verb) are annotated `Voice=Act`

Number	Sing	Plur
Ind	*правене* *pravene*	*правения, правенета* *pravenija, praveneta*
Def	*правенето* *praveneto*	*правенията, правенетата* *pravenijata, pravenetata*

Table 82. [bg] правене (pravene) "doing" NOUN | Aspect=Imp. The rows correspond to different values of Definite. Bulgarian nouns do not inflect for Case.

Number	Sing	Dual	Plur
Nom	дѣлание *dělanije*	дѣлании *dělanii*	дѣланиѣ *dělanija*
Gen	дѣланиѣ *dělanija*	дѣланию *dělaniju*	дѣлании *dělanii*
Dat	дѣланию *dělaniju*	дѣланиема *dělanijema*	дѣланиемъ *dělanijemъ*
Acc	дѣлание *dělanije*	дѣлании *dělanii*	дѣланиѣ *dělanija*
Voc	дѣлание *dělanije*	дѣлании *dělanii*	дѣланиѣ *dělanija*
Loc	дѣлании *dělanii*	дѣланию *dělaniju*	дѣланиихъ *dělaniichъ*
Ins	дѣланиемь *dělanijemь*	дѣланиема *dělanijema*	дѣлании *dělanii*

Table 83. [cu] дѣлание (dělanije) "doing" NOUN | Aspect=Imp. The rows correspond to different values of Case.

| Tense=Imp. All other participles, including some l-participles, are tagged ADJ VerbForm=Part (they actually can take the definite suffix: *миналата, останалите, миналия*). Passive participles have empty Tense. Active participles are distinguished by Tense=Pres (imperfective verbs, progressive meaning) and Tense=Past (the l-participles).

[cu]: All participles are tagged VERB VerbForm=Part and no other part-of-speech tag occurs with the VerbForm feature. Except for the l-participle, which is relatively rare, all participle types can inflect for Case. Active participles are further distinguished by Tense=Pres, Past and in one case even Fut *(*бѫдѫщии*)*. The l-participles have Voice=Act but no Tense; on the other hand, they have currently a special value

of `Aspect=Res`, disregarding the lexical aspect of the lemma. Passive participles use the `Tense` feature to distinguish present and past forms.

19. Conclusion

We have presented the various combinations of morphological features of verbs that occur in Slavic languages, and we have proposed their unified and consistent representation within the Universal Dependencies framework. There already exist UD treebanks of six Slavic languages and we have shown that their authors have not always applied the UD annotation style in the same manner. Datasets for other languages are being prepared at the time of this writing, and their authors will have to take similar decisions. Our proposal should contribute to further harmonization of all these datasets: we hope to trigger discussion that will eventually lead to a more precise specification of UD guidelines for Slavic languages.

Acknowledgments

The author wishes to thank the following people for their valuable comments: Kaja Dobrovoljc, Natalia Kotsyba, Patrice Pognan, Martin Popel, Alexandr Rosen and Zdeněk Žabokrtský. This work has been supported by the Czech Science Foundation (GAČR) grant no. GA15-10472S. It has been using language resources stored and distributed by the LINDAT/CLARIN project of the Ministry of Education, Youth and Sports of the Czech Republic (project LM2015071).

Bibliography

Academia. *Mluvnice češtiny (2) Tvarosloví*. Academia, nakladatelství Československé akademie věd, Praha, Czechoslovakia, 1986.

Breu, Walter. *Probleme der Interaktion von Lexik und Aspekt (ILA)*, volume 412 of *Linguistische Arbeiten*. Niemeyer, Tübingen, Germany, 2000. ISBN 3-484-30412-X.

Comrie, Bernard and Greville G. Corbett. *The Slavonic Languages*. Routledge, London, UK, 2001. ISBN 0-415-04755-2.

Erjavec, Tomaž. MULTEXT-East: Morphosyntactic Resources for Central and Eastern European Languages. *Language Resources and Evaluation*, 46(1):131–142, 2012.

Komárek, Miroslav, Václav Vážný, and František Trávníček. *Historická mluvnice česká II. Tvarosloví*. Státní pedagogické nakladatelství, Praha, Czechoslovakia, 1967.

Nedjalkov, Vladimir P. and Igor' V. Nedjalkov. On the typological characteristics of converbs. In Help, Toomas, editor, *Symposium on language universals*, pages 75–79, Tallinn, Soviet Union, 1987.

Nivre, Joakim, Marie-Catherine de Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajič, Christopher Manning, Ryan McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, Reut Tsarfaty, and Daniel Zeman. Universal Dependencies v1: A Multilingual Treebank Collection. In *Proceedings of the 10th International Conference on Language Resources and Evaluation (LREC 2016)*, Portorož, Slovenia, 2016. European Language Resources Association.

Przepiórkowski, Adam and Marcin Woliński. A Flexemic Tagset for Polish. In *Proceedings of* Morphological Processing of Slavic Languages, *EACL 2003*, 2003. URL `http://nlp.ipipan.waw.pl/~adamp/Papers/2003-eacl-ws12/ws12.pdf`.

Zeman, Daniel. Reusable Tagset Conversion Using Tagset Drivers. In *Proceedings of the 6th International Conference on Language Resources and Evaluation (LREC 2008)*, pages 213–218, Marrakech, Morocco, 2008. European Language Resources Association. ISBN 2-9517408-4-0.

Zeman, Daniel. Slavic Languages in Universal Dependencies. In Gajdošová, Katarína and Adriána Žáková, editors, *Natural Language Processing, Corpus Linguistics, E-learning (proceedings of SLOVKO 2015)*, pages 151–163, Bratislava, Slovakia, 2015. Slovenská akadémia vied, RAM-Verlag. ISBN 978-3-942303-32-3.

Address for correspondence:
Daniel Zeman
`zeman@ufal.mff.cuni.cz`
Ústav formální a aplikované lingvistiky
Matematicko-fyzikální fakulta
Univerzita Karlova v Praze
Malostranské náměstí 25
CZ-11800 Praha, Czechia

The Prague Bulletin of Mathematical Linguistics
NUMBER 105 APRIL 2016

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`.