Dependency parsing representation effects on the accuracy of semantic applications — an example of an inflective language

Lauma Pretkalniņa, Artūrs Znotiņš, Laura Rituma, Didzis Goško
Institute of Mathematics and Computer Science, University of Latvia
Raiņa 29, Rīga, Latvia, LV-1459
E-mail: lauma@ailab.lv, arturs.znotins@gmail.com, laura@ailab.lv, didzis.gosko@gmail.com

Abstract

In this paper we investigate how different dependency representations of a treebank influence the accuracy of the dependency parser trained on this treebank and the impact on several parser applications: named entity recognition, coreference resolution and limited semantic role labeling. For these experiments we use Latvian Treebank, whose native annotation format is dependency based hybrid augmented with phrase-like elements. We explore different representations of coordinations, complex predicates and punctuation mark attachment. Our experiments shows that parsers trained on the variously transformed treebanks vary significantly in their accuracy, but the best-performing parser as measured by attachment score not always leads to best accuracy for an end application.

Keywords: dependency parsing, transformation, extrinsic parser evaluation

1. Introduction

Nowadays syntactic parsers are among the most important language processing resources, but only rarely they are used as standalone applications. Usually syntactic parsers are used as a part of a pipeline for obtaining some semantic information, e.g. named entities, semantic roles, coreferences etc., thus it is important to study how parser properties affect these tools.

Dependency parsers are widely used and often achieve state of the art results (Bohnet and Nivre, 2012), and, thus, are interesting for practical applications. However, while linguists tend to agree, how dependency analysis should be performed on core phenomena, there are several important linguistic phenomena with no consensus available.

In this paper we explore several such phenomena — coordination constructions, punctuation mark attachment and multiword predicates, including compound predicates, compound tense forms and predicates with modality modifiers. We use as data Latvian Treebank where these phenomena are annotated as phrases interconnected with dependency links. We transform these data to pure dependencies varying the annotations of abovementioned phenomena and induce a dependency parser on each of the obtained treebank variants. For purposes of this paper we use MaltParser (Nivre, Kuhlmann, and Hall, 2009) due to the easy availability of implementation for both parser and parameter optimizer MaltOptimizer (Ballesteros and Nivre, 2013), but these results might be generalizable for different dependency parsers as well. We compare accuracy of the obtained parsers as well as the accuracy of the several tools using these parsers. The tools we use for extrinsic evolution are named entity recognizer (NER), coreference resolver (CR) and limited sematic role labeler (SRL).

There are multiple related works featuring extrinsic parser evaluation, yet still exploring the different facets of this problem. (Elmig et al., 2013) compares multiple previously established to-dependency transformations for Penn Treebank. (Johansson and Nugues, 2008) makes comparison between dependency and constituent based parsers for English on SRL, (Miyo, 2008) offers similar comparison on protein-protein interaction. However we provide an insight for an inflective language and for a treebank natively annotated with dependencies in the cases where linguists agree about dependency analysis. Our work is somewhat of a successor to (Nilsson, Nivre, and Hall, 2007) which considers several large European treebanks. While (Schwartz et al., 2011) and (Søgaard, 2013) presents the means to exclude “difficult” phenomena from parser evaluating, we argue that we want parser to annotate these phenomena consistently and in the predefined way, thus, we want to decide how to annotate them before training the parser and then include in the parser evaluation metrics like every other syntactic phenomenon.

The rest of the paper is structured as follows. Section 2 details on Latvian Treebank and the hybrid-to-dependency transformations we used. Section 3 describes the tools we used for extrinsic evaluation (CR, NER, limited SRL). Section 4 provides experimental results. Conclusions and discussion are given in section 5.

2. Treebank and transformations

Latvian is a morphologically rich inflective language with a relatively free word order. Latvian Treebank is being developed since 2010 and currently it contains ~3700 sentences. The treebank is annotated according to the SemTi-Kamols dependency-based grammar model (Pretkalniņa and Rituma, 2013). In essence, each tree is a dependency structure where some nodes are phrases instead of single words. Among constructions annotated as phrases are coordinations, prepositional constructions, appositions, complex predicates etc. This is the way how punctuation marks are linked to the part of sentence invoking their usage, too, as punctuation marks in Latvian can be important disambiguators of the sentence structure.

As several constructions are annotated as phrases, it is possible to choose different ways how to transform these constructions to dependencies when preparing data for dependency parser training. This allows us to prepare different parsers for different semantic tasks.

Hybrid-to-dependency transformations we use are formed as follows.

1. Dependency links connecting two token nodes are left unchanged.
2. Each phrase is transformed to a single-rooted dependency sub-tree connecting all phrase elements. Procedure (rules) for this transformation step is crafted individually for each phrase type.
3. Dependency link with a phrase as a child (or parent) is transformed to dependency link to the root (from the root in case of parent) of sub-tree representing that phrase (obtained by step 2).

In this paper we concentrate on various transformations of three constructions: coordinations, punctuation mark constructs and complex predicates.

Complex predicate (*xpred*) construction is used for annotating compound tense forms (e.g., *[viņš] ir strādājis* '[he] has worked'), compound predicates (e.g., *[viņš] ir skolotājs* '[he] is a teacher') and all kinds of modal constructions (e.g., *[viņš] grib ēst* '[he] wants to eat', *[viņš] varētu būt strādājis [tur]* '[he] might have been working [there]', *[viņš] grib būt skolotājs* '[he] wants to be a teacher'). We test two approaches for such constructions.

1. *BASELEM* — semantically main verb or nominal is chosen as the root of the corresponding dependency sub-tree. Other constituents are made direct children of the root.
2. *DEFAULT* — linearly first constituent, which is not semantically main verb or nominal, is chosen as the root. Other constituents are made direct children of the root.

Punctuation mark construct (*pmc*) construction is used for linking punctuation marks to the word invoking the use of the punctuation marks. It is a phrase-style construction consisting of a base word and all punctuation marks whose usage the base word invokes in this sentence. For example, in Latvian participle clauses are delimited by commas as in *Anna, spēlējot vijoli, neievēroja troksni* 'Anna [while] playing the violin didn't notice the noise'. Here *spēlējot* 'playing'is the dependency head of the participle clause and, thus, invokes both commas, so *spēlējot* and both commas forms punctuation mark construct. We test two approaches for such constructions.

1. *BASELEM* — invocation word is chosen as the root of the corresponding dependency sub-tree. Other constituents are made direct children of the root.
2. *DEFAULT* — linearly first punctuation mark is chosen as the root. Other constituents are made direct children of the root.

Coordination (*coord*) construction is used both for coordinated clauses and coordinated parts of sentence. We test several approaches for such constructions.

1. *3_LEVEL* — first coordinated part is chosen as the root of the corresponding dependency sub-tree. Other coordinated parts are made direct children of the root. Conjuncts and punctuation marks are made direct children of following coordinated part. Hence the name — this structure is three nodes (two dependency links) deep regardless the number of coordinated parts. By classification of (Popel et al., 2013), this coordination annotation approach belongs to Stanford family, fShLsHcFpFdU.
2. *DEFAULT* — conjunction between first two coordinated parts is chosen as root, if there is one, otherwise — punctuation mark between first two coordinated parts. Other constituents are made direct children of the root. This structure is two nodes (one dependency link) deep regardless the number of coordinated parts. By (Popel et al., 2013), this approach belongs to Prague family, fPhLsHcHpBdU.
3. *ROW* — first coordinated part is chosen as root. Each linearly next constituent after the first coordinated part is added as the children of the previous. If there are any conjunctions before first coordinated part, they are made children of the first coordinated part. By (Popel et al., 2013), this approach belongs to Mel'čuk family, fMhLsHcBpBdU.
4. *ROW_NO_CONJ* — first coordinated part is chosen as root. Each linearly next coordinated part is added as the children of the previous. Conjuncts and punctuation marks are made direct children of following coordinated part. By (Popel et al., 2013), this approach belongs to Mel'čuk family, fMhLsHcFpFdU.

Combining transformation choices for each of previously described three constructions, 16 different hybrid-to-dependency transformations for Latvian Treebank are obtained. In following text we will identify these Treebank transformations by stating the transformation type for each of three constructs, e.g., *coordROW & pmcBASELEM & xpredBASELEM*.

Note about dependency roles. When using hybrid-to-dependency transformation on Latvian Treebank, information about phrases in the original mark-up is preserved by decoding it in dependency labels, with the expectation that this would allow to restore parsed sentences back to the richer hybrid representation. Dependency links are augmented with binary flag indicating if dependency parent was originally a phrase or token node. Root node in the sub-tree representing a phrase is labeled with combination of (1) the relation whole phrase carries out in relation to its parent, (2) phrase type, (3) token role within the phrase. Other tokens in sub-tree representing a phrase are labeled with combination of (1) phrase type, (2) token role within the phrase. If an element of the phrase is phrase itself, role element "token role within the phrase" is composite element itself. This yields to a large inventory of roles — several hundreds.

To reduce role inventory, we introduced several measures:

1. Ellipsis is not annotated.
2. Sometimes root node on the sub-tree representing a phrase is labeled only with the relation whole phrase

carries out in relation to its parent.

Shortened rules are used when removing all phrase constituents except root would leave grammatically correct tree E.g., *kaķi un suņi guļ* ‘the cats and the dogs sleep’ — subject here is coordination construction *kaķi un suņi* ‘cats and dogs’. When considering *coordROW* root of the sub-tree representing coordination construction is *kaķi* ‘cats’ and it is labeled only as token dependent subject as *kaķi guļ* is valid sentence. When considering *coordDEFAULT* root of the sub-tree representing coordination construction is *un* ‘and’ and because ‘and’ is not valid standalone subject, it has full labeling: (1) token dependant subject, (2) coordinated parts of sentence, (3) conjunction. Such solution reduces role sparsity without artificially putting under one label totally different tree structures.

These measures reduce role inventory significantly, still we have unconventionally high number of roles in the obtained dependency trees — approx. 200–400 roles depending on transformation (see last column in Table 1).

3. Applications

In order to evaluate and compare parser accuracy on further tasks, we chose several different NLP tasks that use syntactic information as part of their input data, and study the effect of those transformations and parser differences on their accuracy measurements.

3.1 Coreference resolution

Coreference resolution (CR) is the task of finding all expressions that refer to the same discourse entity. LVCoref (Znotins and Paikens, 2014) is a simple rule based coreference resolution system for Latvian. It reaches 66.6% averaged F-score using predicted mentions. Syntactic information provides about 2% increase in averaged F-score compared to flat dependency structure. This contribution may seem rather small, but coreference resolution is mainly based on expression string similarity, syntax provides additional features and constraints.

During mention detection LVCoref uses parser information for noun phrase and their head word (mostly the last word in the phrase) identification.

LVCoref links mentions based on exact string match, precise constructions (appositives, predicative nominatives and acronyms), head matches and pronoun anaphors. Syntactic information is used for appositive (one mention is dependent on another), predicative nominative (mentions are in subject-object relation being dependent on same verb “to be”) and pronoun resolution (antecedents are searched in previous three sentences using classical Hobbs’ algorithm). Syntactic information is important because it creates sentence tree structure leaving attribute words as leaves but head words of noun phrases closer to sentence root. Therefore distance between two mentions (especially main subjects and objects of sentences) should be diminished moving along dependency arcs.

Syntactic information also provides some simple mention compatibility constraints, e.g., two mentions where one dominates another should not be coreferent (excluding appositive construction) — for example *‘[the Supreme Court of [Latvia $_2$] $_1$]’*.

Syntactic features and constraints are applied much more rarely comparing to others so the impact of used parser model should be small.

LVCoref was evaluated against three CR metrics: pairwise, MUC (Vilain et al., 1995) and B^3 (Bagga and Baldwin, 1998). The pairwise metric takes into account all coreferent mention pairs from all predicted and gold coreference chains. MUC is a link based metric which measures how many predicted and gold mention chains need to be merged to cover gold and predicted clusters respectively. B^3 is a mention based metric which measures the proportion of overlap between predicted and gold mention chains for a given mention.

3.2 Named entity recognition

Named Entity Recognition is a well-known natural language processing and Information Extraction task. LVTagger (Paikens et al., 2012; Znotins and Paikens, 2014) is a supervised Named Entity Recognizer (NER) for Latvian, based upon the Stanford NER conditional random field (CRF) classifier (Finkel, Grenager, and Manning, 2005). It recognizes 9 types of named entities (person, location, organization, product, media, profession, sum, time and event) reaching 80.88% F-score for these types.

Typically NER is applied before syntactic parsing because it mainly relies on lexical and gazetteer features. We tried to incorporate a syntactic feature set consisting of

- dependency labels of current word and its local context (two next and three previous words)
- morpho-syntactic information (lemma, part of speech tag, dependency labels) of current word ancestors (moving higher along dependency arcs);
- the closest noun phrase head word morpho-syntactic information, if the current word is included in this phrase.

3.3 Limited semantic role labeling

For semantic role labeling (SRL) experiments we use automatic information extraction system developed for a local news agency (Barzdins et al., 2014). System is predominantly meant for newspaper articles, although the approach is not genre specific. Semantic roles in this approach are understood as frame elements in the theory of semantic frames (Rupenhoffer et al., 2010). The core of SRL system is several decision-tree classifiers trained to identify tokens as frame targets and frame elements. This annotation approach relies on the underlying dependency-tree to automatically derive phrase boundaries once the head-word for the frame target or frame element is selected. Decision-trees for frame target and frame element identification are automatically generated from manually annotated FrameNet-style corpus for Latvian. The corpus contains approx. 5000 sentences from various types of newswire sources. Only 26 Frames which are of interest to the local news-agency are selected for annotation, although this methodology is applicable to any number of frames.

Input text is pre-processed with POS tagger (Paikens et al., 2013), unlabeled dependency parser (Pretkalnina and Rituma, 2013) and NER (Znotins and Paikens, 2014).

SRL system works in two phases. In the first phase, a set of classifiers is used to identify potential target words for each frame type (one classifier for each frame type). In the second phase, another set of classifiers is used to identify frame elements associated with previously identified targets (one classifier for each frame element type). The classifiers use 11 features for frame target identification and 13 features for frame element identification. Some of the features are related to the information of dependency tree — for target recognition the dependency label of potential frame target is used, but for frame element recognition — properties like POS tag and lemma of potential frame element's dependency parent , the path and the distance from the target to the potential frame element. Other features are related to lemma, POS and named entity type for the word currently considered by classifier or for linearly close words to it. Features as path, distance and sometimes roles are affected by differences between annotations given by different parsers, thus, it might provide interesting insight to compare results of SRL system using different parsers.

Annotation differences between parser outputs present an obstacle for the SRL system training and evaluating. The current annotation approach implies that the node annotated as frame element must have all tokens constituting this frame element as its descendants (including itself), and, thus, manually made training data is annotated assuming a fixed syntax model. Because of this, usage of some parsers requires differently annotated training data for SRL system, e.g., if phrase *māsa un brālis* 'sister and brother must be annotated as Relatives then in case of *coordDefault* conjunction *un* must be annotated, but in other *coord* cases — *māsa*. Another example — if phrase *martā tika dibināta SIA "Delta"* 'on March SIA "Delta" was founded' must be annotated as Message then in case of *xpredBASELEM* the main semantic word *dibināta* 'created' must be annotated as the frame element, but in case of *xpredDEFAULT* auxverb *tika* 'was' must be annotated to include all phrase in frame element. However, the frame target annotations are identical for all dependency models, as targets are considered to consist of single token. Due to this we performed the frame target identification phase with all 16 dependency parsers, but the frame element identification phase experiments only with 4 dependency parsers instead of 16. Frame element identification experiments are performed with *coord3_LEVEL & pmcBASELEM & xpredBASELEM*, *coordROW & pmcBASELEM & xpredBASELEM*, and *coordROW_NO_CONJ & pmcBASELEM & xpredBASELEM*, because data originally annotated for *coordROW & pmcBASELEM & xpredBASELEM* is suitable for training these three systems. Assuming that annotating as frame element only the first part of coordinated elements could also be acceptable for some applications, we also include the results of experiment with *coordDEFAULT & pmcBASELEM & xpredBASELEM.*

4. Experiments and results

For all parser-inducing experiments we use Latvian Treebank data transformed to dependency format. Corpus consists of 51946 tokens. Combining all possible values of three transformation parameters (*xpred*, *coord*, *pmc*) we obtain 16 different transformations for Latvian Treebank data and, thus, 16 different data sets for inducing parsers. To get better understanding of the transformation impact on the data, we compared obtained dependency data sets using labeled attachment score (LAS) metric. Two most different annotation pairs are (1) *coordDEFAULT & pmcBASELEM & xpredBASELEM* and *coord3_LEVEL & pmcDEFAULT & xpredDEFAULT,* and (2) *coordROW_NO_CONJ & pmcDEFAULT & xpredDEFAULT* and *coordDEFAULT & pmcBASELEM & xpredBASELEM*. These pairs have only 42.51% equally annotated tokens. The most similar pair is *coordROW_NO_CONJ & pmcBASELEM & xpredBASELEM* and *coord3_LEVEL & pmcBASELEM & xpredBASELEM*, which gives 98.66% LAS. This illustrates that annotation of the phenomena we consider can notably change the machine learning challenge for parser induction.

Each of the 16 parser induction experiments is performed as follows. Data set is tagged by morphological tagger for Latvian (Paikens, Rituma and Pretkalniņa, 2013). 10% of data is held out for final evaluation (the same sentences are chosen for each data set). On the rest of the data optimal MaltParser settings are found with MaltOptimizer (with cross-validation) and a parser is trained. Final results on the test data are given in Table 1 (punctuation is included).

These experiments shows that some of the data sets are notably easier to learn for parser, as unlabeled attachment score (UAS) varies per 6.5 percent points (pp) and labeled attachment score (LAS) — per 7.5 pp. The most easier to learn in terms of LAS is *coordROW_NO_CONJ & pmcDEFAULT & xpredBASELEM* with 67.4%, but in terms of UAS — *coordROW & pmcDEFAULT & xpredDEFAULT* with 75.9%. Results show clear trends about some transformation parameter-value pairs, but not for all:

1. *coordDEFAULT* performs worse than other representations for coordination constructions in terms of LAS and UAS. This comes in lines with conclusion from (Nilsson, Nivre, and Hall, 2007) that Prague family representations are harder to learn.
2. *pmcBASELEM* performs worse than *pmcDEFULT* in terms of LAS.

As our data set has lots of roles, we report label accuracy (LA), too. It interesting to note that highest LA is not yielded neither by model with the smallest role count, nor with the largest. The best LA (74.17%) is given by *coordROW_NO_CONJ & pmcDEFAULT & xpredDEFAULT* with mediocre 293 roles. While (Mille et al., 2012) shows that it gets harder to learn when count of the roles gets much higher than 40, we can't show clear loss of accuracy, comparing approx. 200 and 400 roles. We assume that either there is no such trend for so many roles or that the impact of structural far outweighs the impact of the role count. Also, it is possible that impact of the small size of the data set overshadows expression of such trends.

4.1 Coreference resolution

For coreference experiments we used manually annotated interviews (778 sentences, 13 768 words, 1 088 mentions, and 333 coreference chains) (Znotins and Paikens, 2014). CR experiment results (see Table 2) vary up to 0.94 pp (MUC), 1.24 pp (B^3), 3.60 pp (pairwise) and 1.53 pp (averaged) F-score depending on used parser.

coordDEFAULT parsers lead to the best CR performance regard to all used CR evaluation metrics. *coordROW & pmcDEFAULT* gives comparable results while others lead to worse performance.

Cumulative results of LVCoref show that the impact of used parser for exact string match is insignificant. By adding precise construction rules results varies up to 0.98 pp. Strict head match and pronoun resolution increases these differences up to 1.53 pp. Rules for precise constructions are created considering all transformation types and as we can see there is no obvious correlation between parser results. Results depend on the ability of parser to produce precise syntactic structure needed for specific rule. Last two sets of rules increase differences between parser results mainly because of different syntax tree structure and syntactic constraints.

We argue that *coordDEFAULT* is preferable because it diminishes complex syntax tree structure — all mentions are closer to sentence root and not nested under other coordination parts.

For baseline we used flat dependency structure (arcs between all two proceeding word tokens). Used parsers lead up to 3.95 pp increase in averaged F-score compared to the baseline. Syntactic information is particularly important for pronoun resolution (up to 1.80 pp larger F-score increase over baseline) but also impacts head match (up to 2.09 pp) and precise constructions (up to 0.76 pp).

	LAS (%)	UAS (%)	LA (%)	Role count
coord3_LEVEL & pmcBASELEM & xpredBASELEM	64.32	74.17	72.27	193
coord3_LEVEL & pmcBASELEM & xpredDEFAULT	64.48	75.25	72.86	266
coord3_LEVEL & pmcDEFAULT & xpredBASELEM	66.78	73.75	**74.02**	266
coord3_LEVEL & pmcDEFAULT & xpredDEFAULT	**66.98**	75.17	**73.99**	293
coordDEFAULT & pmcBASELEM & xpredBASELEM	61.02	68.44	70.67	324
coordDEFAULT & pmcBASELEM & xpredDEFAULT	60.85	68.73	69.61	388
coordDEFAULT & pmcDEFAULT & xpredBASELEM	63.61	70.43	70.41	362
coordDEFAULT & pmcDEFAULT & xpredDEFAULT	64.14	71.76	70.12	389
coordROW_NO_CONJ & pmcBASELEM & xpredBASELEM	65.47	75.03	**73.97**	193
coordROW_NO_CONJ & pmcBASELEM & xpredDEFAULT	64.63	**75.61**	72.64	266
coordROW_NO_CONJ & pmcDEFAULT & xpredBASELEM	**67.4**	74.17	**74.13**	266
coordROW_NO_CONJ & pmcDEFAULT & xpredDEFAULT	**67.07**	74.5	**74.17**	293
coordROW & pmcBASELEM & xpredBASELEM	65.07	74.15	72.22	193
coordROW & pmcBASELEM & xpredDEFAULT	65.23	75.1	72.33	266
coordROW & pmcDEFAULT & xpredBASELEM	**67.15**	75.32	73.22	266
coordROW & pmcDEFAULT & xpredDEFAULT	66.36	**75.9**	72.64	293

Table 1: Optimized parser results and role counts for each data set.
Parser results reported in labeled attachment score (LAS), unlabeled attachment score (UAS) and label accuracy (LA).

	MUC (%)	B^3 (%)	Pairwise (%)	AVG$_\text{A}$	AVG$_\text{B}$	AVG$_\text{C}$	AVG$_\text{D}$
Baseline (flat depedency structure)	64.69	75.05	54.38	57.26	58.30	63.79	64.71
coord3_LEVEL & pmcBASELEM & xpredBASELEM	67.78	76.41	58.74	57.76	**59.34**	**66.92**	67.64
coord3_LEVEL & pmcBASELEM & xpredDEFAULT	67.68	76.26	58.25	57.76	59.20	65.93	67.39
coord3_LEVEL & pmcDEFAULT & xpredBASELEM	67.60	76.51	58.80	57.79	59.23	65.99	67.64
coord3_LEVEL & pmcDEFAULT & xpredDEFAULT	67.73	76.60	59.00	57.76	58.85	65.87	67.78
coordDEFAULT & pmcBASELEM & xpredBASELEM	**68.06**	**77.24**	**60.53**	57.76	58.94	66.09	**68.61**
coordDEFAULT & pmcBASELEM & xpredDEFAULT	67.91	76.54	56.95	57.79	59.44	**66.84**	67.13
coordDEFAULT & pmcDEFAULT & xpredBASELEM	**68.16**	**77.43**	60.13	57.84	59.14	65.85	**68.57**
coordDEFAULT & pmcDEFAULT & xpredDEFAULT	**68.02**	**77.40**	**60.55**	57.76	**59.30**	**66.75**	**68.66**
coordROW_NO_CONJ & pmcBASELEM & xpredBASELEM	67.72	76.34	58.77	57.82	**59.37**	**66.81**	67.61
coordROW_NO_CONJ & pmcBASELEM & xpredDEFAULT	67.58	76.19	58.35	57.82	59.00	66.01	67.37
coordROW_NO_CONJ & pmcDEFAULT & xpredBASELEM	67.22	76.26	58.74	57.79	59.17	66.06	67.41
coordROW_NO_CONJ & pmcDEFAULT & xpredDEFAULT	67.50	76.46	59.13	57.76	58.64	65.64	67.70
coordROW & pmcBASELEM & xpredBASELEM	67.87	76.37	58.44	57.76	58.89	66.15	67.56
coordROW & pmcBASELEM & xpredDEFAULT	**68.11**	76.77	58.02	57.76	**59.27**	65.92	67.63
coordROW & pmcDEFAULT & xpredBASELEM	67.88	**77.15**	59.90	57.82	**59.62**	66.17	**68.31**
coordROW & pmcDEFAULT & xpredDEFAULT	67.92	**77.20**	**60.38**	57.74	59.23	**66.59**	**68.50**

Table 2: Coreference experiment results reported in F-score for MUC, B^3, and pairwise evaluation metrics and cumulative averaged metric score as rule sets are added: exact string match (AVG$_\text{A}$), precise constructions (AVG$_\text{B}$), strict head match (AVG$_\text{C}$) and finally pronoun anaphora (AVG$_\text{D}$).

	F1 (%)	Precision (%)	Recall (%)
Baseline (no syntactic features used)	80.88	85.12	77.04
coord3_LEVEL & pmcBASELEM & xpredBASELEM	**81.46**	85.70	**77.63**
coord3_LEVEL & pmcBASELEM & xpredDEFAULT	81.21	85.26	77.52
coord3_LEVEL & pmcDEFAULT & xpredBASELEM	80.96	85.38	76.97
coord3_LEVEL & pmcDEFAULT & xpredDEFAULT	81.12	85.61	77.08
coordDEFAULT & pmcBASELEM & xpredBASELEM	81.11	85.05	77.52
coordDEFAULT & pmcBASELEM & xpredDEFAULT	**81.63**	**86.21**	77.52
coordDEFAULT & pmcDEFAULT & xpredBASELEM	81.14	85.80	76.97
coordDEFAULT & pmcDEFAULT & xpredDEFAULT	81.07	85.78	76.85
coordROW_NO_CONJ & pmcBASELEM & xpredBASELEM	**81.53**	85.71	**77.74**
coordROW_NO_CONJ & pmcBASELEM & xpredDEFAULT	80.51	84.80	76.63
coordROW_NO_CONJ & pmcDEFAULT & xpredBASELEM	80.98	85.29	77.08
coordROW_NO_CONJ & pmcDEFAULT & xpredDEFAULT	80.93	85.47	76.85
coordROW & pmcBASELEM & xpredBASELEM	80.93	85.19	77.08
coordROW & pmcBASELEM & xpredDEFAULT	**81.46**	85.70	**77.63**
coordROW & pmcDEFAULT & xpredBASELEM	**81.45**	**86.07**	77.30
coordROW & pmcDEFAULT & xpredDEFAULT	81.28	85.84	77.19

Table 3: NER experiment results.

	F1 (%)	Precision (%)	Recall (%)	LA (%)
Baseline (no syntactic features used)	**60.9**	**68.5**	54.8	—
coord3_LEVEL & pmcBASELEM & xpredBASELEM	**60.4**	60.4	**60.4**	72.27
coord3_LEVEL & pmcBASELEM & xpredDEFAULT	59.7	**61.7**	57.8	72.86
coord3_LEVEL & pmcDEFAULT & xpredBASELEM	57.9	58.8	57	74.02
coord3_LEVEL & pmcDEFAULT & xpredDEFAULT	58.5	59.3	57.8	73.99
coordDEFAULT & pmcBASELEM & xpredBASELEM	59.8	58.5	**61.1**	70.67
coordDEFAULT & pmcBASELEM & xpredDEFAULT	59.7	59.9	59.6	69.61
coordDEFAULT & pmcDEFAULT & xpredBASELEM	**60.5**	**61.5**	59.6	70.41
coordDEFAULT & pmcDEFAULT & xpredDEFAULT	60.2	60.8	59.6	70.12
coordROW_NO_CONJ & pmcBASELEM & xpredBASELEM	58.9	58.9	58.9	73.97
coordROW_NO_CONJ & pmcBASELEM & xpredDEFAULT	59.5	59.7	59.3	72.64
coordROW_NO_CONJ & pmcDEFAULT & xpredBASELEM	57	57	57	74.13
coordROW_NO_CONJ & pmcDEFAULT & xpredDEFAULT	59.1	60.2	58.1	74.17
coordROW & pmcBASELEM & xpredBASELEM	58.3	58.1	58.5	72.22
coordROW & pmcBASELEM & xpredDEFAULT	58.5	58	58.9	72.33
coordROW & pmcDEFAULT & xpredBASELEM	57.4	57.4	57.4	73.22
coordROW & pmcDEFAULT & xpredDEFAULT	60	60.8	59.3	72.64

Table 4: SRL frame target identification results. Label accuracy (LA) of parsers repeated for comparison.

	F1 (%)	Precision (%)	Recall (%)
Baseline (no syntactic features used)	53.4	58.5	49.1
coord3_LEVEL & pmcBASELEM & xpredBASELEM	67.7	63.1	72.9
coordROW_NO_CONJ & pmcBASELEM & xpredBASELEM	**71.7**	**68.8**	**74.9**
coordROW & pmcBASELEM & xpredBASELEM	68.4	64.8	72.3
coordDEFAULT & pmcBASELEM & xpredBASELEM	68.1	64.6	72.1

Table 5: SRL frame element identification results for selected parsers.

4.2 Named entity recognition

For NER experiments we used manually annotated corpus (2500 sentences, 45 000 words) that consists of news articles. NER experiment results (see Table 3) shows that there is no clear winner among all 16 parsers. Results vary from 80.51% to 81.63% F-score (84.80–86.21% for precision and 77.74–76.63% for recall). *coordDEFAULT & pmcBASELEM & xpredDEFAULT* gives the best F-score (81.63%) and precision (86.21%). *coordROW_NO_CONJ & pmcBASELEM & xpredBASELEM* parser reaches the second best result while performance of other *coordROW_NO_CONJ* parsers are much worse. *coordROW* parsers seems to provide more stable results with good overall performance.

The used syntactic feature set gives 0.75 pp F-score increase (1.09 pp precision and 0.70 pp recall) considering parser that produces the best performance for NER.

4.3 Limited semantic role labeling

SRL system is trained and tested on manually annotated news-wire texts. Training data contains 4445 sentences, 2255 of them with non-empty frames, and 4746 non-empty frames. Test data contains 478 sentences, 141 of them with non-empty frames, and 270 non-empty frames. For these experiments parser-independent NER was used.

Frame target identification results are shown in Table 4. Results vary up to 3 pp in F-score and they show no notable correlation with parser label accuracy scores and do not seem to favor any particular kind for dependency transformations. When compared to baseline using syntactic features tends to notably raise the recall and lower the precision while slightly lowering F-score. The best results in terms of F-score are obtained with *coordDEFAULT & pmcDEFAULT & xpredBASELEM* and *coord3_LEVEL & pmcBASELEM & xpredBASELEM* parsers. Their F-score is 0.4–0.5 pp lower than baseline, but their recall is up to 4.8 pp higher.

Frame element identification experiment results (see Table 5) vary up to 4 pp in F-score and even 5.7 pp in precision. The best performance is given by *coordROW_NO_CONJ & pmcBASELEM & xpredBASELEM* system. Even though *coordROW_NO_CONJ & pmcBASELEM & xpredBASELEM* parser has slightly better results than the other three parsers we doubt that alone is responsible for performance increase for SRL system. Thus, we are lead to believe that this type of coordination representation — *coordROW_NO_CONJ* — is more appropriate for frame element identification. However, experiments with bigger frame corpus could be desirable.

However, the frame target identification results for *coordROW_NO_CONJ* is noteworthy 2 pp below the best frame target identification system. This leads to thinking that total performance of the SRL system can be improved by either not using syntactic information for target identification or if better target identification recall is important for SLR application combining each of SRL systems steps with the parser most appropriate for it as some parsers are more useful for frame element identification and some — for frame target identification.

Considering the SRL frame element identification system with *coordDEFAULT & pmcBASELEM & xpredBASELEM* parser it is important to note that despite relatively low parser accuracy, SRL system gives competitive performance increase compared to baseline system with no syntactic features. Thus, despite that *coordDEFAULT* schemes are harder to learn for parser, they are easier to use for some tools.

5. Conclusion

In this paper we show more evidence that dependency parser accuracy alone is not enough to judge its suitability for various parser applications. We also see that different dependency annotation schemes should be chosen depending on the target applications. We examine 16 parsers trained on different representations of a single corpus Latvian Treebank) and evaluate three information extraction tools — coreference resolver (CR), named entity recognizer (NER), and limited FrameNet-style semantic role labeler (SRL) — based on these parsers. The examined parsers differ in representations of coordinations, complex predicates and punctuation mark attachment. SRL system consists of two parts: frame target identification and frame element identification.

Parser results in terms of LAS varies up to 6.6 pp and CR results — 4 pp (full system, see AVG$_D$ in Table 2). The smallest effect from the dependency annotation differences has NER — F-score varies up to 1.1 pp. SRL frame target identification results in terms of F-sore varies up to 3.5 pp. Using syntactic features for target identification does not rise F-score, but it notably rises recall. Frame element identification results comparing 4 parsers in terms of F-sore — up to 4 pp.

While frame element identification system prefers *coordROW_NO_CONJ & pmcBASELEM & xpredBASELEM* parser, frame target identification system prefers *coordDEFAULT & pmcDEFAULT & xpredBASELEM* parser, thus, leading to think that the best overall performance for SRL system could be obtained by using different parser for each task or using no syntactic features for target identification — depending on what is deemed to be more important: target identification precision or recall.

Parser results acknowledge that Prague style coordination (*coordDEFAULT*) representation is notably harder to learn for a parser (leading to 2.2–3.6 pp LAS decrease), however, CR gives the best results with these parsers. Also, NER and SRL systems with *coordDEFAULT* parsers give comparably good results.

In future work we would like to investigate further the interaction between frame element identification task and other types of dependency annotations. Also, as the manually annotated corpora used here are rather small for some tasks, similar larger scale experiments also will be useful.

6. Acknowledgements

The research leading to these results has received funding from the research project “Information and Communication Technology Competence Center” of EU Structural funds, contract nr. L-KC-11-0003, signed between ICT Competence Centre and Investment and Development Agency of Latvia, Research No. 2.7 “Creation of the New Information Archive Access Product based on Advanced NLP” and was partially supported by Latvian 2010–2014 National Research Program No. 2 project No. 5.

We also thank the University of Latvia for the financial support in preparing and presenting this paper and the anonymous reviewers for their improvement recommendations.

7. References

Bagga, A. and Baldwin, B. (1998). *Algorithms for scoring coreference chains.* Proceedings of the First International Conference on Language Resources and Evaluation Workshop on Linguistics Coreference, Granada, Spain, pp. 563–566

Ballesteros, M. and Nivre, J. (2012). *MaltOptimizer: An Optimization Tool for MaltParser.* Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2012), Stroudsburg, PA, USA: Association for Computational Linguistics, pp. 58–62.

Barzdins, G., Gosko, D., Rituma, R., and Paikens, P. (2014). *Using C5.0 and Exhaustive Search for Boosting Frame-Semantic Parsing Accuracy*. Proceedings of the 9th Language Resources and Evaluation Conference (LREC 2014). Reykjavik, Iceland, *this volume*.

Bohnet, B. and Nivre, J. (2012). *A Transition-Based System for Joint Part-of-Speech Tagging and Labeled*

Non-Projective Dependency Parsing. Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, Stroudsburg, PA, USA: Association for Computational Linguistics, pp. 1455–1465.

Elming, J., Johannsen, A., Klerke, S., Lapponi, E., Martinez, H., and Søgaard, A. (2013). *Down-stream effects of tree-to-dependency conversions.* Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2013), Stroudsburg, PA, USA: Association for Computational Linguistics, pp. 617–626.

Finkel, J. R., Grenager, T., and Manning, C. (2005). *Incorporating Non-local Information into Information Extraction Systems by Gibbs Sampling*. Proceedings of the 43nd Annual Meeting of the Association for Computational Linguistics (ACL-05), Stroudsburg, PA, USA: Association for Computational Linguistics pp. 363–370.

Johansson, R. and Nugues, P. (2008). *The Effect of Syntactic Representation on Semantic Role Labeling.* Proceedings of the 22nd International Conference on Computational Linguistics (COLING 2008), Vol. 1, Stroudsburg, PA, USA: Association for Computational Linguistics, pp. 393–400

Mille, S., Burga, A., Ferraro, G., and Wanner, L. (2012). *How Does the Granularity of an Annotation Scheme Influence Dependency Parsing Performance?* Proceedings of the 24nd International Conference on Computational Linguistics (COLING 20112): Posters, Mumbai, India: The COLING 2012 Organizing Committee, pp. 839–852.

Miyao, Y., Sætre, R., Sagae, K., Matsuzaki; T., and Tsujii; J. (2008). *Task-oriented Evaluation of Syntactic Parsers and Their Representations.* Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL-08:HLT), Stroudsburg, PA, USA: Association for Computational Linguistics, pp. 46–54.

Nilsson, J., Nivre, J., and Hall, J. (2007). *Generalizing Tree Transformations for Inductive Dependency Parsing.* Proceedings of 45th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL-07:HLT), Stroudsburg, PA, USA: Association for Computational Linguistics, pp. 968–975.

Nivre, J., Kuhlmann, M., and Hall, J. (2009). *An Improved Oracle for Dependency Parsing with Online Reordering.* Proceedings of the 11th International Conference on Parsing Technologies (IWPT 2009), Stroudsburg, PA, USA: Association for Computational Linguistics, pp. 73–76.

Paikens, P., Auzina, I., Garkaje, G., and Paegle, M. (2012). *Towards named entity annotation of Latvian National Library corpus.* Proceedings of the 5th International Conference on Human Language Technologies — the Baltic Perspective, Frontiers in Artificial Intelligence and Applications, Vol. 247, IOS Press, pp. 169–175.

Paikens, P., Rituma, L., and Pretkalniņa, L. (2013). *Morphological analysis with limited resources: Latvian example.* Proceedings of the 19th Nordic Conference of Computational Linguistics (NODALIDA 2013), NEALT Proceedings Series, Vol. 16, Linköping, Sweden: Linköping University Electronic Press, pp. 267–277.

Popel, M., Mareček, D., Štěpánek, J., Zeman, D., and Žabokrtský, Z. (2013). *Coordination Structures in Dependency Treebanks.* Proceedings of the 51th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL-13:HLT), Vol. 1, Stroudsburg, PA, USA: Association for Computational Linguistics, pp. 517–527.

Pretkalniņa, L., Rituma, L. (2013). *Statistical syntactic parsing for Latvian.* Proceedings of the 19th Nordic Conference of Computational Linguistics (NODALIDA 2013), NEALT Proceedings Series, Vol. 16, Linköping, Sweden: Linköping University Electronic Press, pp. 279–289.

Ruppenhofer, J., Ellsworth, M., Petruck, M.R.L., Johnson, C.R., and Scheffczyk, J. (2010). *FrameNet II: Extended Theory and Practice*, Berkeley, CA, USA: International Computer Science Institute.

Schwartz, R., Abend, O., Reichart, R., and Rappoport, A. (2011). *Neutralizing Linguistically Problematic Annotations in Unsupervised Dependency Parsing Evaluation.* Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL-11:HLT), Vol. 1, Stroudsburg, PA, USA: Association for Computational Linguistics, pp. 663–672.

Søgaard, A. (2013). An empirical study of differences between conversion schemes and annotation guidelines. Proceedings of the Second International Conference on Dependency Linguistics (DepLing 2013), Prague, Czech Republic: Charles University in Prague, Matfyzpress, pp. 298–307.

Vilain, M., Burger, J., Aberdeen, J., Connolly, D., and Hirschman, L. (1995). *A Model-theoretic Coreference Scoring Scheme* Proceedings of the 6th Conference on Message Understanding (MUC-95), Stroudsburg, PA, USA: Association for Computational Linguistics, pp. 45–52.

Znotins, A. and Paikens, P. (2014). *Coreference Resolution for Latvian.* Proceedings of the 9th Language Resources and Evaluation Conference (LREC 2014). Reykjavik, Iceland, *this volume*.