A Persian Treebank with Stanford Typed Dependencies

Mojgan Seraji, Carina Jahani, Beáta Megyesi, Joakim Nivre

Uppsala University, Department of Linguistics and Philology
E-mail: firstname.lastname@lingfil.uu.se

Abstract

We present the Uppsala Persian Dependency Treebank (UPDT) with a syntactic annotation scheme based on Stanford Typed Dependencies. The treebank consists of 6,000 sentences and 151,671 tokens with an average sentence length of 25 words. The data is from different genres, including newspaper articles and fiction, as well as technical descriptions and texts about culture and art, taken from the open source Uppsala Persian Corpus (UPC). The syntactic annotation scheme is extended for Persian to include all syntactic relations that could not be covered by the primary scheme developed for English. In addition, we present open source tools for automatic analysis of Persian containing a text normalizer, a sentence segmenter and tokenizer, a part-of-speech tagger, and a parser. The treebank and the parser have been developed simultaneously in a bootstrapping procedure. The result of a parsing experiment shows an overall labeled attachment score of 82.05% and an unlabeled attachment score of 85.29%. The treebank is freely available as an open source resource.

Keywords: Treebank, Persian, Stanford Typed Dependencies

1. Introduction

During the past years many language resources have been developed for different languages including syntactically annotated corpora, *treebanks*. Treebanks play an important role in developing applications involving natural language parsing as well as in empirical linguistic studies. Treebanks are based on different grammatical theories. In recent years more interest has been drawn towards developing treebanks with dependency-based annotation, which is well suited for languages with flexible word order. The Stanford Typed Dependencies (STD) representation (De Marneffe and Manning, 2008) is a dependency-based structure that was originally developed for English but it has been designed to be cross-linguistically valid and is based on a set of universal grammatical relations. So far, the scheme has successfully been adapted to various languages representing different types of languages, such as Chinese (Chang et al., 2009), Finnish (Haverinen et al., 2010), and Modern Hebrew (Tsarfaty, 2013).

Until recently, Persian despite its large number of speakers in the world (over 100 million) still belonged to the group of languages with less developed linguistically annotated data sets. Lately, we have witnessed the emergence of three treebanks, namely, the HPSG-based PerTreeBank (Ghayoomi, 2012), the Uppsala Persian Dependency Treebank (Seraji et al., 2012b), and the Persian Treebank (Rasooli et al., 2013). The development of the Uppsala Persian Dependency Treebank and the HPSG-based treebank started almost simultaneously in different places with different annotation schemes. Shortly after, the Persian Treebank was developed in Iran with the annotation scheme based on traditional Persian grammar.

With respect to the linguistic properties and the relatively high degree of free word order in Persian, we opted for a dependency-based structure using the Stanford Typed Dependencies scheme. Earlier work on the creation of the treebank and the tools have been presented in Seraji et al. (2012a). In this paper, we present the syntactic dependency annotation of the final version of the Uppsala Persian Dependency Treebank, and tools that have been developed for the morpho-syntactic annotation of Persian. First, we will give a description of the annotation scheme, followed by a description of the tools used in the treebank development.

2. The Treebank

The Uppsala Persian Dependency Treebank (UPDT)[1] (Seraji et al., 2013) is a syntactically annotated corpus of contemporary Persian based on dependency grammar. The treebank consists of 6,000 annotated and validated sentences, 151,671 word tokens, and 15,692 word types. The average sentence length in the treebank is 25 words. The treebank data was taken from the open source, validated Uppsala Persian Corpus[2] (UPC) (Seraji et al., 2012a). This corpus is a modified version of the Bijankhan corpus (Bijankhan, 2004) and is currently the largest freely available corpus of Persian, created from on-line texts with manually validated linguistic annotation. UPC differs from the original version concerning the standardization of orthography, added sentence segmentation, more consistent tokenization and morphological annotation. The entire corpus consists of 2,703,265 tokens, annotated with part-of-speech tags and morpho-syntactic and partly semantic features.

We extracted the first 6,000 sentences of UPC to serve as our treebank data. The data is from different genres, including newspaper articles and fiction, as well as technical descriptions and texts about culture and art. The treebank is open source and freely available in CoNLL-format.[3] A comprehensive description of the extended version of Stanford Typed Dependencies for Persian and the morphosyntactic features can be found in the Uppsala Persian Dependency Treebank Annotation Guidelines (Seraji et al., 2013).

[1] http://stp.lingfil.uu.se/∼mojgan/UPDT.html
[2] http://stp.lingfil.uu.se/∼mojgan/UPC.html
[3] http://ilk.uvt.nl/conll/#dataformat

We use a syntactic annotation scheme based on dependency structure, where each dependency relation is annotated with a functional category, indicating the grammatical function of the dependent to the head. The annotation scheme is based on Stanford Typed Dependencies (De Marneffe et al., 2006), which has become a de facto standard for English. The dependency annotation of a sentence always forms a tree representing all tokens of the sentence (including punctuation marks) and rooted at an artificial root node prefixed to the sentence. Thus, we adopt the so-called *basic* version of STD (with punctuation retained), as opposed to the *collapsed* version, where some tokens may not correspond to nodes in the dependency structure and a single node may have more than one incoming arc. In general, every token in a sentence is assigned a syntactic head and one dependency label. Table 1 lists all atomic labels used in the syntactic annotation of UPDT, with new relations in italics.

While we have tried to keep the label and construction set as close as possible to the original scheme, we have extended the scheme in order to include all syntactic relations that could not be covered by the primary scheme developed for English. Altogether we have added 10 new relations to describe various relations in light verb constructions *acomp-lvc*, *dobj-lvc*, *nsubj-lvc*, *prep-lvc*, the accusative marker *rā acc*, object of comparative *cpobj*, comparative modifier *cprep*, topic dependent *dep-top*, vocative dependent *dep-voc*, and foreign word *fw*. For instance, *dobj-lvc* denotes a direct object functioning as the nominal part in a complex predicate (light verb construction), as illustrated in Figure 1, where the complex predicate تأثیر می‌گیرند (effect take-3pl-pres = are affected) consists of the verb می‌گیرند (take-3pl-pres) and the nominal part تأثیر (effect). The extended version of STD for Persian has a total of 101 dependency relations of which 48 (including 10 new additions) are used for indicating basic relations. The remaining 53 labels are complex, and are used to assign syntactic relations to words containing unsegmented clitics.

In the case of complex unsegmented word forms, we use complex labels where the first label indicates the main syntactic function while subsequent labels mark other functions carried by elements incorporated in the word form. The additional functions are listed in the order in which they occur and are prefixed with a backward slash (\) if they precede the segment carrying the main function and a forward slash (/) if they follow it. Thus, the label *poss/pc* is assigned to a word that has the main function *poss* and an additional (enclitic) *pc* element. By contrast, the label *ccomp\poss* is used for (the head of) a clausal complement, which is itself enclitic on a *poss* element. In Table 1, we only list atomic labels. A complete list of all simple and complex labels (with frequency information) can be found in to the Uppsala Persian Dependency Treebank Annotation Guidelines (Seraji et al., 2013).

Some dependency relations from the original STD scheme have been excluded in the Persian STD since the corresponding relations either do not exist or are not applicable in Persian. For instance, we have not found any use of the dependency relation indirect object (*iobj*). Indirect objects are always realized as prepositional phrases in Persian, so the relations prepositional modifier (*prep*) and prepositional object (*pobj*) are sufficient. The excluded relations in the extended version of Persian STD are: abbreviation modifier (*abbrev*), agent (*agent*), attributive (*attr*), clausal subject (*csubj*), clausal passive subject (*csubjpass*), expletive (*expl*), infinitival modifier (*infmod*),

Category	Description
acc	*Accusative marker*
acomp	Adjectival complement
acomp-lvc	*Adjectival complement in light verb construction*
advcl	Adverbial clause modifier
advmod	Adverbial modifier
amod	Adjectival modifier
appos	Appositional modifier
aux	Auxiliary
auxpass	Passive auxiliary
cc	Coordination
ccomp	Clausal complement
complm	Complementizer
conj	Conjunct
cop	Copula
cpobj	*Object of comparative*
cprep	*Comparative modifier*
dep	Dependent
dep-top	*Topic Dependent*
dep-voc	*Vocative Dependent*
det	Determiner
dobj	Direct object
dobj-lvc	*Direct object in light verb construction*
fw	*foreign word*
mark	Marker
mwe	Multi-word expression
neg	Negation modifier
nn	Noun compound modifier
npadvmod	Nominal adverbial modifier
nsubj	Nominal subject
nsubj-lvc	*Nominal subject in light verb construction*
nsubjpass	Passive nominal subject
num	Numeric modifier
number	Element of compound number
parataxis	Parataxis
pobj	Object of a preposition
poss	Possession modifier
predet	Predeterminer
prep	Prepositional modifier
prep-lvc	*Prepositional modifier in light verb construction*
prt	Phrasal verb particle
punct	Punctuation
quantmod	Quantifier phrase modifier
rcmod	Relative clause modifier
rel	Relative
root	Root
tmod	Temporal modifier
xcomp	Open clausal complement

Table 1: Syntactic relations in UPDT with new relations in italics.

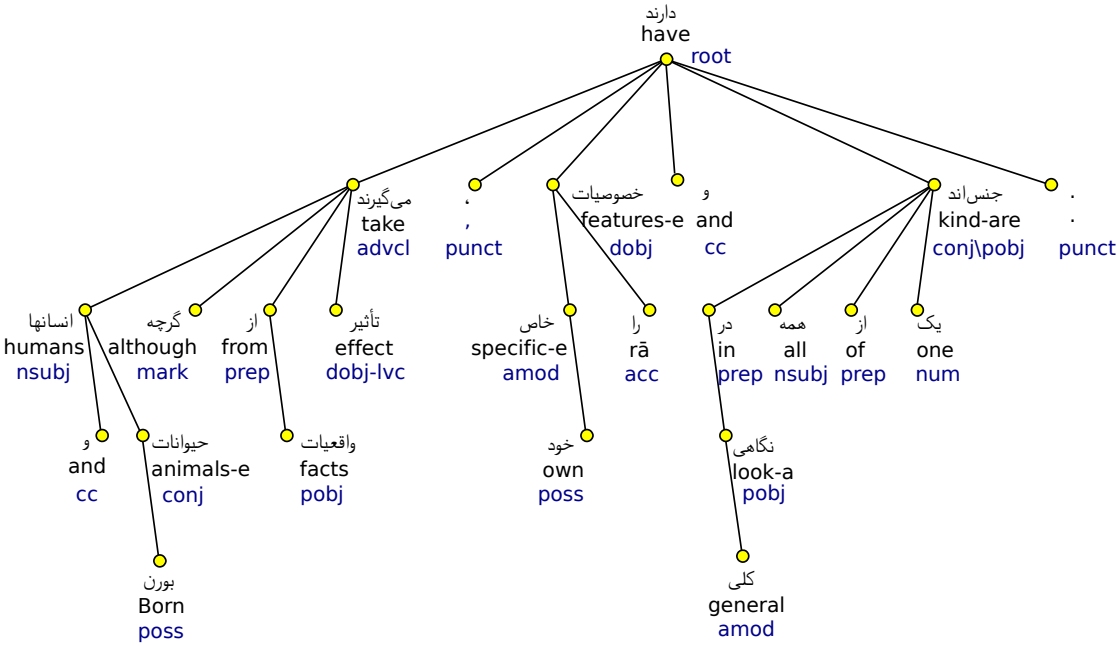

Figure 1: Syntactic annotation for a Persian sentence.
Gloss: Humans and animals-e Born although from facts effect take, features-e specific-e own -rā have-3pl-pres and in look-a general all of one sort-are.
Translation: Although (Adolf) Born's humans and animals are affected by circumstances, they have their own special characteristics and in general all are of the same kind.

participial modifier (*partmod*), prepositional complement (*pcomp*), possessive modifier (*possessive*), and purpose clause modifier (*purpcl*).

Figure 1 shows the dependency annotation for a sentence from UPDT about the Czech artist *Adolf Born*, with English glosses. The sentence consists of the subordinate clause *humans and animals-e*[4] *Born although from facts effect take-3pl-pres* (Although Born's humans and animals are affected by circumstances) and the main clause *features-e specific-e own -rā have and in look-a general all of one sort-are* (they have their own special characteristics and in general all are of the same kind). The subordinate clause is an adverbial clause modifier with the root *take-3pl-pres* marked by the label *advcl* and governing the nominal subjects *humans and animals-e Born*, the subordinate conjunction *although*, the prepositional modifier *from* followed by the prepositional object *facts*, and the preverbal direct object *effect* in light verb construction with *take-3pl-pres*. The nominal subjects *humans* and *animals-e* are coordinated and linked with an ezāfe construction to their possessive modifier *Born*. The main clause is rooted at the verb *have-3pl-pres* which governs an implied subject,[5] the direct object *features-e specific-e own -rā*, the coordinating conjunction *and*, and the coordinated verb phrase *in look-a general all of one sort-are*. The direct object is headed by *features-e*, which is linked by an ezāfe construction to its adjectival modifier *specific-e* and further to its genitive complement *own*. The direct object contains additionally the accusative marker *-rā*. The coordinated verb phrase *sort-are* governs the prepositional modifier *in look-a general*, the nominal subject *all*, and the second prepositional modifier *of*. The first prepositional modifier is rooted at the preposition *in* linked to its object *look-a*[6] which is modified by the adjectival modifier *general*. The second prepositional modifier *of* has its object *sort* in the form of complex element with the attached copula clitic ـاند /-and/ (be-3pl-pres) modified with the numeric modifier *num*. Thus the coordinated verb *sort-are* has received the complex label *conj\pobj*. In other words, the *conj* (conjunct) is itself enclitic on a *pobj* (prepositional object) element. Since we gave priority to the verb to be the important part in the syntactic structure and the verb is attached to the preposition object, the preposition object which should actually be under the *prep* ends up higher in the structure.

[4] An ezāfe /e/ is an unstressed enclitic particle that links the elements within a noun phrase, adjective phrase and prepositional phrase indicating the semantic relation between the phrasal elements and is represented by the short vowel /e/ after consonants or /ye/ after vowels.

[5] The subject is absent (pro-drop) but the information is given by the verb through the attached personal ending ـند /-and/ (3pl).

[6] The indefinite marker ی /i/ (a, an), as enclitic particle, is joined to the noun *look*.

3. The Tools

Before creating the UPDT, we have used and modified available tools for Persian in order to improve the quality of Bijankhan corpus into UPC. The goal was to make the corpus more appropriate for syntactic annotation. Therefore, for each step of processing, from normalization to syntactic parsing, there is a developed tool. Figure 2 shows a pipeline containing a chain of tools for automatic analysis of Persian text. Each tool will be briefly presented in the following subsections.

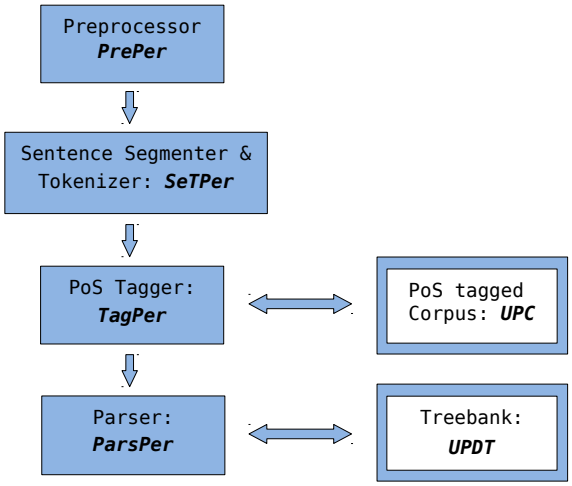

Figure 2: Persian natural language processing pipeline.

3.1. Preprocessing

One of the major bottlenecks of automatic processing of Persian is the lack of standardization in Persian orthography in terms of different writing styles, spacing and font encoding. Persian orthography is not consistent. One word may be written in various forms and with different Unicode characters in a text. Compound words and inflectional affixes are highly affected and can be written either as attached to or detached from their adjacent word (cf. Seraji et al., 2012a). These inconsistencies can easily impact the tokenization process which in turn affects the quality of morphological and syntactic analysis. Therefore in creating the UPDT, we normalized our data using the open source software program *PrePer*[7] (Seraji et al., 2012a) that was developed for editing and cleaning up texts in Persian to solve the inconsistency issues. The software uses the *Virastar* module (Bargi, 2011) for some formatting tasks.

3.2. Sentence Segmentation and Tokenization

In UPC, sentences are separated by one of the punctuation marks '.', '!', '؟', or combinations thereof. In addition, the punctuation mark ':' has been treated as a sentence separator when used to introduce a list of alternatives. Tokenization in UPC has been made more consistent, compared to the original corpus, by treating all words separated by whitespace or punctuation as separate tokens, except in cases where single words with internal white space can be identified deterministically and unambiguously. White space (by *PrePer*) has been replaced with zero-width non-joiner to make sure that tokens in the treebank never contain internal whitespace. Clitics attached to their head words without whitespace have not been separated from their heads but are given a special analysis in the syntactic annotation instead. For normalizing the sentence segmentation and tokenization of Persian texts, the open source *SeTPer*[8] (Seraji et al., 2012a) was developed. SeTPer was created through reusing and modifying the sentence segmenter and tokenizer tools in the modular software platform Uplug, a system designed for the integration of text processing tools (Tiedemann, 2003).

3.3. Morphological Annotation and Tagging

The morphological annotation in UPC consists of 32 part-of-speech tags that encode a subset of the features found in the original Bijankhan corpus. The tag set is listed with explanations in Table 2. There are 15 main part-of-speech categories consisting of adjective, adverb, clitic, conjunction, delimiter, determiner, foreign word, interjection, symbol, noun, numeral, preposition, preverbal particle, pronoun, and verb. In addition, categories such as adjective, adverb, noun, and verb are annotated for morphological and some semantic features. A part-of-speech tagger, named TagPer, was developed when the statistical tagger HunPoS (Halácsy et al., 2007) was trained on UPC (Seraji et al., 2012a). TagPer resulted in an overall accuracy of 97.4% for Persian when HunPoS was trained on 90% of the data and evaluated on the remaining 10%. TagPer is used as a freely available tool for part-of-speech tagging of Persian.[9]

3.4. Syntactic Annotation and Parsing

In order to syntactically annotate the sentences we employed MaltParser (Nivre et al., 2006) in a bootstrapping scenario. We started by training MaltParser on a small seed sample of manually annotated sentences and used the induced model to parse the rest of the corpus. We selected a subset of the parsed sentences for manual correction, added them to the training set, retrained the parser, and reparsed the remaining corpus. The process was iterated as the size of the treebank grew and the quality of the parser improved. The development of the parser and the treebank have been accomplished simultaneously and the quality of the parser has been improved steadily.

In order to annotate and correct our syntactic annotation in a tree structure we used the free software TrEd tree editor.[10] TrEd (Hajič et al., 2001) is a fully programmable and customizable graphical user interface for tree-like structures and was used as the main annotation tool for the Prague Dependency Treebank. From TrEd we export annotations in the CoNLL-X format (Buchholz and Marsi,

[7] http://stp.lingfil.uu.se/∼mojgan/preper.html

[8] http://stp.lingfil.uu.se/∼mojgan/setper.html

[9] http://stp.lingfil.uu.se/∼mojgan/tagper.html

[10] TrEd is licensed under the GNU General Public License and is available at http://ufal.mff.cuni.cz/∼pajas/.

Category	Description
ADJ	Adjective
ADJ_CMPR	Comparative adjective
ADJ_INO	Participle adjective
ADJ_SUP	Superlative adjective
ADJ_VOC	Vocative adjective
ADV	Adverb
ADV_COMP	Adverb of comparison
ADV_I	Adverb of interrogation
ADV_LOC	Adverb of location
ADV_NEG	Adverb of negation
ADV_TIME	Adverb of time
CLITIC	Accusative marker
CON	Conjunction
DELM	Delimiter
DET	Determiner
FW	Foreign Word
INT	Interjection
N_PL	Plural noun
N_SING	Singular noun
NUM	Numeral
N_VOC	Vocative noun
P	Preposition
PREV	Preverbal particle
PRO	Pronoun
SYM	Symbol
V_AUX	Auxiliary verb
V_COP	Verb copula
V_IMP	Imperative verb
V_PA	Past tense verb
V_PP	Past participle verb
V_PRS	Present tense verb
V_SUB	Subjunctive verb

Table 2: Part-of-speech tags in UPC.

2006), which is the official distribution format of UPDT.

For parser evaluation, the treebank has been split sequentially into 10 parts, of which segments 1–8 are used for training (80%), 9 for development (10%) and 10 for final testing (10%). To evaluate the performance of MaltParser trained on the treebank, we first tuned parameters using the development set and then retrained the parser on the combined training and development set (90%). The final evaluation resulted in a labeled attachment score 82.05% and an unlabeled attachment score of 85.29%. The developed parser *ParsPer* is a freely available tool for syntactic parsing of Persian.[11]

4. Conclusion

We presented the open source Uppsala Persian Dependency Treebank, containing 6,000 sentences and 151,671 tokens, with the annotation scheme based on Stanford Typed Dependencies. In addition we presented freely available tools for the automatic processing of Persian, namely, tools for normalizing, sentence segmenting and tokenizing, part-of-speech tagging, and parsing.

[11] http://stp.lingfil.uu.se/∼mojgan/parsper.html

5. References

Bargi, Alan Aziz. (2011). Virastar. https://github.com/aziz/virastar.

Bijankhan, Mahmood. (2004). The role of the corpus in writing a grammar: An introduction to a software. *Iranian Journal of Linguistics*, 19.

Buchholz, Sabine and Marsi, Erwin. (2006). CoNLL-X shared task on multilingual dependency parsing. In *Proceedings of the Tenth Conference on Computational Natural Language Learning (CoNLL)*, pages 149–164.

Chang, Pi-Chuan, Tseng, Huihsin, Jurafsky, Dan, and Manning, Christopher D. (2009). Discriminative reordering with Chinese grammatical relations features. In *Proceedings of the Third Workshop on Syntax and Structure in Statistical Translation (SSST-3) at NAACL HLT 2009*, pages 51–59.

De Marneffe, Marie-Catherine and Manning, Christopher D. (2008). The Stanford Typed Dependencies Representation. In *Proceedings of the COLING'08 Workshop on Cross-Framework and Cross-Domain Parser Evaluation*, pages 1–8.

De Marneffe, Marie-Catherine, MacCartney, Bill, and Manning, Christopher D. (2006). Generating Typed Dependency Parses from Phrase Structure Parses. In *Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC)*, pages 449–454.

Ghayoomi, Masood. (2012). Bootstrapping the Development of an HPSG-based Treebank for Persian. *Journal of Linguistic Issues in Language Technology*, 7:105–114.

Hajič, Jan, Hladká, Barbora Vidová, and Pajas, Petr. (2001). Prague Dependency Treebank: Annotation Structure and Support. In *Proceeding of the IRCS Workshop on Linguistic Databases, Philadelphia*, pages 105–114.

Halácsy, Péter, Kornai, András, and Oravecz, Csaba. (2007). HunPos: an Open Source Trigram Tagger. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Interactive Poster and Demonstration Sessions (ACL)*, pages 209–212.

Haverinen, Katri, Viljanen, Timo, Laippala, Veronika, Kohonen, Samuel, Ginter, Filip, and Salakoski, Tapio. (2010). Treebanking Finnish. In *Proceedings of the Ninth International Workshop on Treebanks and Linguistic Theories (TLT)*, pages 79–90.

Nivre, Joakim, Hall, Johan, and Nilsson, Jens. (2006). Maltparser: A Data-Driven Parser-Generator for Dependency Parsing. In *Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC)*, pages 2216–2219.

Rasooli, Mohammad Sadegh, Kouhestani, Manouchehr, and Moloodi, Amirsaeid. (2013). Development of a Persian Syntactic Dependency Treebank. In *Proceedings of North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 306–314.

Seraji, Mojgan, Megyesi, Beáta, and Nivre, Joakim. (2012a). A Basic Language Resource Kit for Persian. In *Proceedings of the 8th International Conference on Lan-*

guage Resources and Evaluation (LREC), pages 2245–2252.

Seraji, Mojgan, Megyesi, Beáta, and Nivre, Joakim. (2012b). Bootstrapping a Persian Dependency Treebank. *Linguistic Issues in Language Technology*, 7(18):1–10.

Seraji, Mojgan, Jahani, Carina, Megyesi, Beáta, and Nivre, Joakim. (2013). The Uppsala Persian Dependency Treebank Annotation Guidelines. Technical report, Department of Linguistics and Philology, Uppsala University.

Tiedemann, Jörg. (2003). *Recycling Translation - Extraction of Lexical Data from Parallel Corpora and their Application in Natural Language Processing*. PhD Thesis, Studia Linguistica Upsaliensia 1.

Tsarfaty, Reut. (2013). A Unified Morpho-Syntactic Scheme of Stanford Dependencies. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 578–584.