A Unified Morpho-Syntactic Scheme of Stanford Dependencies

Reut Tsarfaty
Uppsala University, Sweden
tsarfaty@stp.lingfil.uu.se

Abstract

Stanford Dependencies (SD) provide a functional characterization of the grammatical relations in syntactic parse-trees. The SD representation is useful for parser evaluation, for downstream applications, and, ultimately, for natural language understanding, however, the design of SD focuses on structurally-marked relations and under-represents morphosyntactic realization patterns observed in Morphologically Rich Languages (MRLs). We present a novel extension of SD, called Unified-SD (U-SD), which unifies the annotation of structurally- and morphologically-marked relations via an inheritance hierarchy. We create a new resource composed of U-SD-annotated constituency and dependency treebanks for the MRL Modern Hebrew, and present two systems that can automatically predict U-SD annotations, for gold segmented input as well as raw texts, with high baseline accuracy.

1 Introduction

Stanford Dependencies (SD) provide a functional characterization of the grammatical relations in syntactic trees, capturing the predicate-argument structure of natural language sentences (de Marneffe et al., 2006). The SD representation proved useful in a range of downstream tasks, including Textual Entailments (Dagan et al., 2006) and BioNLP (Fundel and Zimmer., 2007), and in recent years SD structures have also become a de-facto standard for parser evaluation in English (de Marneffe and Manning, 2008a; Cer et al., 2010; Nivre et al., 2010). Efforts now commence towards extending SD for cross-lingual annotation and evaluation (McDonald et al., 2013; Che et al., 2012; Haverinen et al., 2011). By and large, these efforts aim to remain as close as possible to the original SD scheme. However, the original SD design emphasizes word-tokens and configurational structures, and consequently, these schemes overlook properties and realization patterns observed in a range of languages known as *Morphologically Rich Languages (MRLs)* (Tsarfaty et al., 2010).

MRLs use word-level affixes to express grammatical relations that are typically indicated by structural positions in English. By virtue of word-level morphological marking, word-order in MRLs may be flexible. MRLs have been a focal point for the parsing community due to the challenges that these phenomena pose for systems originally developed for English.[1] Here we argue that the SD hierarchy and design principles similarly emphasize English-like structures and under-represent morphosyntactic argument-marking alternatives. We define an extension of SD, called Unified-SD (U-SD), which unifies the annotation of structurally and morphologically marked relations via an inheritance hierarchy. We extend SD with a functional branch, and provide a principled treatment of morpho-syntactic argument marking.

Based on the U-SD scheme we create a new parallel resource for the MRL Modern Hebrew, whereby aligned constituency and dependency trees reflect equivalent U-SD annotations (cf. Rambow (2010)) for the same set of sentences. We present two systems that can automatically learn U-SD annotations, from the dependency and the constituency versions respectively, delivering high baseline accuracy on the prediction task.

[1] See also the SPMRL line of workshops https://sites.google.com/site/spsemmrl2012/ and the MT-MRL workshop http://cl.haifa.ac.il/MT/.

2 The Challenge: SD for MRLs

Stanford Dependencies (SD) (de Marneffe et al., 2006; de Marneffe and Manning, 2008b) deliver a functional representation of natural language sentences, inspired by theoretical linguistic work such as studies on Relational Grammars (Postal and Perlmutter, 1977), Lexical Functional Grammars (LFG) (Bresnan, 2000) and the PARC dependency scheme (King et al., 2003). At the same time, the scheme is designed with end-users in mind, allowing them to utilize parser output in a form which is intuitively interpretable and easily processed.

SD basic trees represent sentences as binary relations between word tokens. These relations are labeled using traditional grammatical concepts (*subject, object, modifier*) that are arranged into an inheritance hierarchy (de Marneffe and Manning, 2008a, Sec. 3). There are different versions of SD annotations: the basic SD scheme, which annotates surface dependency relations as a tree spanning all word tokens in the sentence, and the collapsed SD version, in which function words (such as prepositions) are collapsed and used for specifying a direct relation between content words.

The SD scheme defines a core set of labels and principles which are assumed to be useful for different languages. However, a close examination of the SD label-set and inheritance hierarchy reveals that some of its design principles are geared towards English-like (that is, configurational) phenomena, and conflict with basic properties of MRLs. Let us list three such design principles and outline the challenges that they pose.

2.1. SD relate input-tokens. In MRLs, substantial information is expressed as word affixes. One or more morphemes may be appended to a content word, and several morphemes may be contained in a single space-delimited token. For example, the Hebrew token *wkfraiti*[2] in (1) includes the morphemes *w* (and), *kf* (when) and *raiti* (saw); the latter segment is a content word, and the former two are functional morphemes.

(1) *wkfraiti* *at*
and-when-saw.1st.Singular acc
hsrj *hifn*
the-movie the-old
w/and-1.1 *kf*/when-1.2 *raiti*/saw-1.3 *at*/acc-2 *h*/the-3.1 *srj*/movie-3.2 *h*/the-4.1 *ifn*/old-4.2

[2]We use the transliteration of Sima'an et al. (2001).

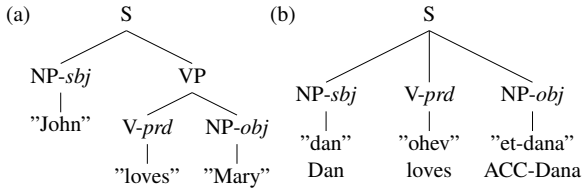

Figure 1: English (a) and Hebrew (b) PS trees decorated with function labels as dash features.

Naïvely taking input tokens as words fails to capture meaningful relations between morphological segments internal to space-delimited tokens.

2.2. SD label structurally-marked relations. Configurational languages like English use function words such as prepositions and auxiliaries to indicate relations between content words and to mark properties of complete structures. In MRLs, such relations and properties may be indicated by word-level morphological marking such as case (Blake, 1994) and agreement (Corbett, 2006). In (1), for instance, the case marker *at* indicates an *accusative* object relation between "see" and "movie", to be distinguished from, e.g, a *dative* object. Moreover, the agreement in (1) on the *definite* morpheme signals that "old" modifies "movie". While the original SD scheme label-set covers function words (e.g. *auxpass, expl, prep*), it misses labels for bound morphemes that mark grammatical relations across languages (such as *accusative, dative* or *genitive*). Explicit labeling of such relational morphemes will allow us to benefit from the information that they provide.

2.3. SD relations may be inferred using structural cues. SD relations are extracted from different types of trees for the purpose of, e.g., cross-framework evaluation (Cer et al., 2010). Insofar, recovering SD relations from phrase-structure (PS) trees have used a range of structural cues such as positions and phrase-labels (see, for instance, the software of de Marneffe and Manning (2008a)). In MRLs, positions and phrase types may not suffice for recovering SD relations: an NP under S in Hebrew, for instance, may be a *subject* or an *object*, as shown in Figure 1, and morphological information then determines the function of these constituents. Automatically inferring predicate-argument structures across treebanks thus must rely on both structural and morphological marking, calling for a single annotation scheme that inter-relate the marking alternatives.

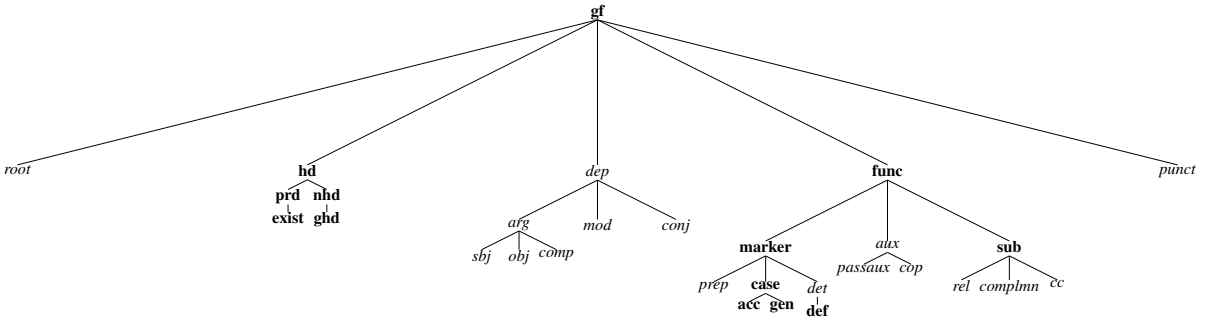

Figure 3: **The Unified SD (U-SD) Ontology.** The architectural changes from the original SD scheme: (i) added a *hd* branch, for implicit head labels; (ii) added a *func* branch where all functional elements (*prep, aux, cc, rel*) as well as morphological markers are moved under; (iii) there is a clear separation between open-class categories (which fall under *hd*, *dep*), closed class elements (under *func*) and non-words (*root* and *punct*). **Boldface** elements are new to U-SD. *Italic* branches spell out further as in the original SD.

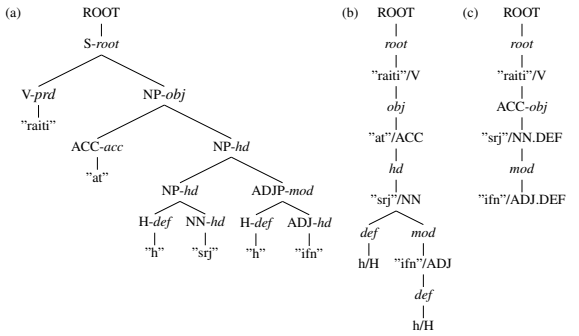

Figure 2: **Sample U-SD Trees** for sentence (1). (a) a phrase-structure tree decorated with U-SD labels, (b) a basic U-SD tree, and (c) a collapsed U-SD tree, where functional nodes are consumed.

3 The Proposal: Unified-SD (U-SD)

To address these challenges, we propose an extension of SD called Unified-SD (U-SD) which annotates relations between morphological segments and reflects different types of argument-marking patterns. The SD ontology is re-organized and extended to allow us to annotate morphologically- and structurally-marked relations alike.

Preliminaries. We assume that $\mathcal{M}(w_1...w_n) = s_1....s_m$ is a morphological analysis function that identifies all morphological segments of a sentence $S = w_1...w_n$. The U-SD scheme provides the syntactic representation of S by means of a set of triplets (l, s_i, s_j) consisting of a label l, a head s_i and a dependent s_j $(i \neq j)$. The segments are assumed to be numbered x.y where x is the position of the input token, and y is the position of the segment inside the token. The segmentation numbering is demonstrated in Example (1).

The U-SD Hierarchy. Figure 3 shows our proposed U-SD hierarchy. Everything in the ontology is of type *gf* (grammatical function). We define five ontological sub-types: *root, hd, dep, func, punct*. The *root* marks a special root dependency. The *dep* branch is used for dependent types, and it retains much of the structure in the original SD scheme (separating *sbj* types, *obj* types, *mod* types, etc.). The new *func* branch contains argument-marking elements, that is, function words and morphemes that play a role in indicating properties or grammatical relations in the syntactic representation. These functional elements may be of types *marker* (prepositions and case), *aux* (auxiliary verbs and copular elements) and *sub* (subordination/conjunction markers). All inherited *func* elements may be consumed (henceforth, collapsed) in order to infer grammatical properties and relations between content words. Head types are implicit in dependency triplets, however, when decorating PS trees with dependency labels as dash features or edge features (as in TigerXML formats (Brants et al., 2002) or via unification-based formalisms) both heads and dependents are labeled with their grammatical types (see Figure 2(a)). The *hd* branch extends the scheme with an inventory of argument-taking elements, to be used when employing SD inside constituency treebanks. The *punct* branch is reserved for punctuation, prosody and other non-verbal speech acts. The complete ontology is given in the appendix.

Annotation Guidelines. Anderson (1992) delineates three kinds of properties that are realized by morphology: *structural, inherent*, and *agreement* properties. Structural properties (e.g., case) are marked on a content word to indicate its rela-

Gold:		Segments	Functions
	DEP	1.00	0.8475
	RR	1.00	**0.8984**

Predicted:		Segments	Functions
	DEP	1.00	0.8349
	RR	1.00	**0.8559**

Raw:		Segments	Functions
	DEP	0.9506	0.7817
	RR	0.9603	**0.8130**

Table 1: Inferring U-SD trees using different frameworks. All numbers report labeled TedEval accuracy.

tion to other parts of the sentence. Inherent properties (gender, number, etc.) indicate inherent semantic properties of nominals. Agreement properties indicate the semantic properties of nominals on top of other elements (verbs, adjectives, etc.), in order to indicate their relation to the nominals.

We define annotation guidelines that reflect these different properties. Structural morphemes (case) connect words in the arc-structure, linking a head to its semantic dependent, like the case marker "at"-ACC in Figure 2(b). Inherent / agreement properties are annotated as dependents of the content word they add properties to, for instance, the prefixes *def* in Figure 2(b) hang under the modified noun and adjective.

Collapsed U-SD structures interpret *func* elements in order to refine the representation of relations between content words. Case markers can be used for refining the relation between the content words they connect by labeling their direct relation, much like *prep* in the original SD scheme (see, e.g., the ACC-*obj* in Figure 2c). Inherent/agreement features are in fact features of their respective head word (as the X.DEF nodes in Figure 2c).[3] Auxiliaries may further be used to add tense/aspect to the main predicate, and subordinators may propagate information inside the structure (much like conjunction is propagated in SD).

Universal Aspects of U-SD. The revised U-SD ontology provides a typological inventory of labels that describe different types of arguments (*dep*), argument-taking elements (*hd*), and argument-marking elements (*func*) in the grammar of different languages. Abstract (universal) concepts reside high in the hierarchy, and more specific distinctions, e.g., morphological markers of particular types, are daughters within more specific branches. Using U-SD for evaluating monolingual parsers is best done with the complete label set relevant for that language. For cross-language evaluation, we can limit the depth of the hierarchy, and convert the more specific notions to their most-specific ancestor in the evaluation set.

[3]Technically, this is done by deleting a line adding a property to the morphology column in the CoNLL format.

4 Automatic Annotation of U-SD Trees

Can U-SD structures be automatically predicted? For MRLs, this requires disambiguating both morphological and syntactic information. Here we employ the U-SD scheme for annotating morphosyntactic structures in Modern Hebrew, and use these resources to train two systems that predict U-SD annotations for raw texts.[4]

Data. We use the Modern Hebrew treebank (Sima'an et al., 2001), a corpus of 6220 sentences morphologically segmented and syntactically analyzed as PS trees. We infer the function label of each node in the PS trees based on the morphological features, syntactic environment, and dash-feature (if exist), using deterministic grammar rules (Glinert, 1989). Specifically, we compare each edge with a set of templates, and, once finding a template that fits the morphological and syntactic profile of an edge, we assign functions to all daughters. This delivers PS trees where each node is annotated with a U-SD label (Figure 2a). At a second stage we project the inferred labels onto the arcs of the unlabeled dependency trees of Goldberg (2011), using the tree unification operation of Tsarfaty et al. (2012a). The result is a dependency tree aligned with the constituency tree where dependency arcs are labeled with the same function as the respective span in the PS tree.[5]

Systems. We present two systems that predict U-SD labels along with morphological and syntactic information, using **[DEP]**, a dependency parser (Nivre et al., 2007), and **[RR]**, a Relational-Realizational (RR) constituency parser (Tsarfaty and Sima'an, 2008). DEP is trained directly on the dependency version of the U-SD resource. Since it cannot predict its own segmentation, automatic segments and tags are predicted using the system of Adler and Elhadad (2006). The constituency-

[4]Despite significant advances in parsing Hebrew, as of yet there has been no functional evaluation of Hebrew parsers. E.g., Goldberg and Elhadad (2010) evaluate on unlabeled dependencies, Tsarfaty (2010) evaluate on constituents. This is largely due to the lack of standard resources and guidelines for annotating functional structures in such a language.

[5]The resources can be downloaded at `http://www.tsarfaty.com/heb-sd/`.

based model is trained on U-SD-labeled RR trees using Petrov et al. (2006). We use the lattice-based extension of Goldberg and Elhadad (2011) to perform joint segmentation and parsing. We evaluate three input scenarios: **[Gold]** gold segmentation and gold tags, **[Predicted]** gold segments, and **[Raw]** raw words. We evaluate parsing results with respect to basic U-SD trees, for 42 labels. We use TedEval for joint segmentation-tree evaluation (Tsarfaty et al., 2012b) and follow the cross-parser evaluation protocol of Tsarfaty et al. (2012a).

Results. Since this work focuses on creating a new resource, we report results on the standard devset (Table 1). The gold input scenarios obtain higher accuracy on function labels in all cases, since gold morphological analysis delivers disambiguated functions almost for free. Constituency-based RR structures obtain better accuracy on U-SD annotations than the respective dependency parser. All in all, the U-SD seed we created allows us to infer rich interpretable annotations automatically for raw text, using either a dependency parser or a constituency parser, in good accuracy.

5 Conclusion

The contribution of this paper is three-fold. We offer a principled treatment of annotating MRLs via a Unified-SD scheme, which we design to be applicable to many languages. We deliver new U-SD annotated resources for the MRL Modern Hebrew, in different formal types. We finally present two systems that automatically predict U-SD annotations for raw texts. These structures are intended to serve semantic applications. We further intend to use this scheme and computational frameworks to serve a wide cross-parser investigation on inferring functional structures across languages.

Appendix: The U-SD Ontology

The list in (2) presents the complete U-SD ontology. The hierarchy employs and extends the SD label set of de Marneffe et al. (2006). For readability, we omit here various compound types under *mod*, including *nn*, *mwe*, *predet* and *preconj*.

(2) *gf* *root* - root
- *hd* - head (governor, argument-taking)
  - *prd* - verbal predicate
    - *exist* - head of an existential phrase
  - *nhd* - head of a nominal phrase
    - *ghd* - genitive head of a nominal phrase
- *dep* - dependent (governed, or an argument)
  - *arg* - argument
    - *agent* - agent
    - *comp* - complement
      - *acomp* - adjectival complement
      - *ccomp* - comp clause with internal sbj
      - *xcomp* - comp clause with external sbj
      - *pcomp* - comp clause of a preposition
    - *obj* - object
      - *dobj* - direct object
      - *gobj* - genitive object
      - *iobj* - indirect object
      - *pobj* - object of a preposition
    - *subj* - subject
      - *expl* - expletive subject
      - *nsubj* - nominal subject
      - — *nsubjpass* - passive nominal sbj
      - *csubj* - clausal subject
      - — *csubjpass* - passive clausal sbj
  - *mod* - modifier
    - *appos* - apposition/parenthetical
    - *abbrev* - abbreviation
    - *amod* - adjectival modifier
    - *advmod* - adverbial modifier
    - — *neg* - negative modifier
    - *prepmod* - prepositional modifier
    - — *possmod* - possession modifier
    - — *tmod* - temporal modifier
    - *rcmod* - relative clause modifier
    - *infmod* - infinitival modifier
    - *nummod* - numerical modifier
  - *parataxis* - "side-by-side", interjection
  - *conj* - conjuct
- *func* - functional (argument marking)
  - *marker* - nominal-marking elements
    - *prep* - preposition
    - *case* - case marker
    - — *acc* - accusative case
    - — *dat* - dative case
    - — *gen* - genitive case
    - — *nom* - nominative case
    - *det* - determiner
    - — *def* - definite marker
    - — *dem* - demonstrative
  - *sub* - phrase-marking elements
    - *complm* - introducing comp phrase
    - *rel* - introducing relative phrase
    - *cc* - introducing conjunction
    - *mark* - introducing an advb phrase
  - *aux* - auxiliary verb or a feature-bundle
    - *auxpass* - passive auxiliary
    - *cop* - copular element
    - *modal* - modal verb
    - *qaux* - question auxiliary
- *punct* - punctuation

Acknowledgements

We thank Joakim Nivre, Yoav Goldberg, Djamé Seddah and anonymous reviewers for comments and discussion. This research was partially funded by the Swedish Research Council. The author is now a researcher at the Weizmann Institute.

References

Meni Adler and Michael Elhadad. 2006. An unsupervised morpheme-based HMM for Hebrew morphological disambiguation. In *Proceedings of COLING-ACL*.

Stephen R. Anderson. 1992. *A-Morphous Morphology*. Cambridge University Press.

Barry J. Blake. 1994. *Case*. Cambridge University Press, Cambridge.

Sabine Brants, Stefanie Dipper, Silvia Hansen, Wolfgang Lezius, and George Smith. 2002. The TIGER treebank. In *Proceedings of TLT*.

Joan Bresnan. 2000. *Lexical-Functional Syntax*. Blackwell.

Daniel Cer, Marie-Catherine de Marneffe, Daniel Jurafsky, and Christopher D. Manning. 2010. Parsing to stanford dependencies: Trade-offs between speed and accuracy. In *Proceedings of LREC*.

Wanxiang Che, Valentin I. Spitkovsky, and Ting Liu. 2012. A comparison of chinese parsers for stanford dependencies. In *Proceedings of ACL*, pages 11–16.

Greville G. Corbett. 2006. *Agreement*. Cambridge Textbooks in Linguistics. Cambridge University Press.

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2006. The PASCAL recognising textual entailment challenge. In *MLCW 2005, LNAI Volume*.

Marie-Catherine de Marneffe and Christopher D. Manning. 2008a. Stanford dependencies manual. Technical Report.

Marie-Catherine de Marneffe and Christopher D. Manning. 2008b. The stanford typed dependencies representation. In *Proceedings of the workshop on Cross-Framework and Cross-Domain Parser Evaluation*.

Marie-Catherine de Marneffe, Bill MacCartney, and Christopher D. Manning. 2006. Generating typed dependency parses from phrase structure parses. In *Proceedings of LREC*, pages 449–454.

Robert Kuffner Fundel, Katrin and Ralf Zimmer. 2007. RelEx relation extraction using dependency parse trees. *Bioinformatics*, (23).

Lewis Glinert. 1989. *The Grammar of Modern Hebrew*. Cambridge University Press.

Yoav Goldberg and Michael Elhadad. 2010. Easy-first dependency parsing of Modern Hebrew. In *Proceedings of NAACL/HLT workshop on Statistical Parsing of Morphologically Rich Languages*.

Yoav Goldberg and Michael Elhadad. 2011. Joint Hebrew segmentation and parsing using a PCFGLA lattice parser. In *Proceedings of ACL*.

Yoav Goldberg. 2011. *Automatic syntactic processing of Modern Hebrew*. Ph.D. thesis, Ben Gurion University of the Negev.

Katri Haverinen, Filip Ginter, Samuel Kohonen, Timo Viljanen, Jenna Nyblom, and Tapio Salakoski. 2011. A dependency-based analysis of treebank annotation errors. In *Proceedings of DepLing*.

Tracy H. King, Richard Crouch, Stefan Riezler, Mary Dalrymple, and Ronald Kaplan. 2003. The PARC 700 dependency bank. In *The 4th International Workshop on Linguistically Interpreted Corpora*.

Ryan McDonald, Joakim Nivre, Yvonne Quirmbach-Brundage, Yoav Goldberg, Dipanjan Das, Kuzman Ganchev, Keith Hall, Slav Petrov, Hao Zhang, Oscar Täckström, Claudia Bedini, Núria Bertomeu Castelló, and Jungmee Lee. 2013. Universal dependency annotation for multilingual parsing. In *Proceedings of ACL*.

Joakim Nivre, Jens Nilsson, Johan Hall, Atanas Chanev, Gülsen Eryigit, Sandra Kübler, Svetoslav Marinov, and Erwin Marsi. 2007. Maltparser: A language-independent system for data-driven dependency parsing. *Natural Language Engineering*, 13(1):1–41.

Joakim Nivre, Laura Rimell, Ryan McDonald, and Carlos Gómez-Rodríguez. 2010. Evaluation of dependency parsers on unbounded dependencies. In *Proceedings of COLING*, pages 813–821.

Slav Petrov, Leon Barrett, Romain Thibaux, and Dan Klein. 2006. Learning accurate, compact, and interpretable tree annotation. In *Proceedings of ACL*.

Paul M. Postal and David M. Perlmutter. 1977. Toward a universal characterization of passivization. In *BLS 3*.

Owen Rambow. 2010. The Simple Truth about Dependency and Phrase Structure Representations: An Opinion Piece. In *Proceedings of HLT-ACL*.

Khalil Sima'an, Alon Itai, Yoad Winter, Alon Altman, and Noa Nativ. 2001. Building a Tree-Bank for Modern Hebrew Text. In *Traitement Automatique des Langues*.

Reut Tsarfaty and Khalil Sima'an. 2008. Relational-realizational parsing. In *Proceedings of CoLing*.

Reut Tsarfaty, Djame Seddah, Yoav Goldberg, Sandra Kuebler, Marie Candito, Jennifer Foster, Yannick Versley, Ines Rehbein, and Lamia Tounsi. 2010. Statistical parsing for morphologically rich language (SPMRL): What, how and whither. In *Proceedings of the first workshop on Statistical Parsing of Morphologically Rich Languages (SPMRL) at NA-ACL*.

Reut Tsarfaty, Joakim Nivre, and Evelina Andersson. 2012a. Cross-framework evaluation for statistical parsing. In *Proceeding of EACL*.

Reut Tsarfaty, Joakim Nivre, and Evelina Andersson. 2012b. Joint evaluation for segmentation and parsing. In *Proceedings of ACL*.

Reut Tsarfaty. 2010. *Relational-Realizational Parsing*. Ph.D. thesis, University of Amsterdam.