Down-stream effects of tree-to-dependency conversions

Jakob Elming, Anders Johannsen, Sigrid Klerke, Emanuele Lapponi[†], Hector Martinez, Anders Søgaard
Center for Language Technology, University of Copenhagen
[†]Institute for Informatics, University of Oslo

Abstract

Dependency analysis relies on morphosyntactic evidence, as well as semantic evidence. In some cases, however, morphosyntactic evidence seems to be in conflict with semantic evidence. For this reason dependency grammar theories, annotation guidelines and tree-to-dependency conversion schemes often differ in how they analyze various syntactic constructions. Most experiments for which constituent-based treebanks such as the Penn Treebank are converted into dependency treebanks rely blindly on one of four-five widely used tree-to-dependency conversion schemes. This paper evaluates the down-stream effect of choice of conversion scheme, showing that it has dramatic impact on end results.

1 Introduction

Annotation guidelines used in modern dependency treebanks and tree-to-dependency conversion schemes for converting constituent-based treebanks into dependency treebanks are typically based on a specific dependency grammar theory, such as the Prague School's Functional Generative Description, Meaning-Text Theory, or Hudson's Word Grammar. In practice most parsers constrain dependency structures to be tree-like structures such that each word has a single syntactic head, limiting diversity between annotation a bit; but while many dependency treebanks taking this format agree on how to analyze many syntactic constructions, there are still many constructions these treebanks analyze differently. See Figure 1 for a standard overview of clear and more difficult cases.

The difficult cases in Figure 1 are difficult for the following reason. In the easy cases morphosyntactic and semantic evidence cohere. Verbs govern subjects morpho-syntactically and seem semantically more important. In the difficult cases, however, morpho-syntactic evidence is *in conflict* with the semantic evidence. While auxiliary verbs have the same distribution as finite verbs in head position and share morpho-syntactic properties with them, and govern the infinite main verbs, main verbs seem semantically superior, expressing the main predicate. There may be distributional evidence that complementizers head verbs syntactically, but the verbs seem more important from a semantic point of view.

Tree-to-dependency conversion schemes used to convert constituent-based treebanks into dependency-based ones also take different stands on the difficult cases. In this paper we consider four different conversion schemes: the Yamada-Matsumoto conversion scheme **yamada**,[1] the CoNLL 2007 format **conll07**,[2] the conversion scheme **ewt** used in the English Web Treebank (Petrov and McDonald, 2012),[3] and the **lth** conversion scheme (Johansson

[1]The Yamada-Matsumoto scheme can be replicated by running penn2malt.jar available at http://w3.msi.vxu.se/∼nivre/research/Penn2Malt.html. We used Malt dependency labels (see website). The Yamada-Matsumoto scheme is an elaboration of the Collins scheme (Collins, 1999), which is not included in our experiments.

[2]The CoNLL 2007 conversion scheme can be obtained by running pennconverter.jar available at http://nlp.cs.lth.se/software/treebank_converter/with the 'conll07' flag set.

[3]The EWT conversion scheme can be replicated using the Stanford converter available at http://nlp.stanford.edu/software/stanford-dependencies.shtml

Proceedings of NAACL-HLT 2013, pages 617–626,
Atlanta, Georgia, 9–14 June 2013.

Clear cases		Difficult cases	
Head	Dependent	?	?
Verb	Subject	Auxiliary	Main verb
Verb	Object	Complementizer	Verb
Noun	Attribute	Coordinator	Conjuncts
Verb	Adverbial	Preposition	Nominal
			Punctuation

Figure 1: Clear and difficult cases in dependency annotation.

and Nugues, 2007).[4] We list the differences in Figure 2. An example of differences in analysis is presented in Figure 3.

In order to access the impact of these conversion schemes on down-stream performance, we need extrinsic rather than intrinsic evaluation. In general it is important to remember that while researchers developing learning algorithms for part-of-speech (POS) tagging and dependency parsing seem obsessed with accuracies, POS sequences or dependency structures have no interest on their own. The accuracies reported in the literature are only interesting insofar they correlate with the usefulness of the structures predicted by our systems. Fortunately, POS sequences and dependency structures *are* useful in many applications. When we consider tree-to-dependency conversion schemes, down-stream evaluation becomes particularly important since some schemes are more fine-grained than others, leading to lower performance as measured by intrinsic evaluation metrics.

Approach in this work

In our experiments below we apply a state-of-the-art parser to five different natural language processing (NLP) tasks where syntactic features are known to be effective: negation resolution, semantic role labeling (SRL), statistical machine translation (SMT), sentence compression and perspective classification. In all five tasks we use the four tree-to-dependency conversion schemes mentioned above and evaluate them in terms of down-stream performance. We also compare our systems to baseline systems not relying on syntactic features, when possible, and to results in the literature, when comparable results exist. Note that negation resolution and SRL are not end applications. It is not easy to generalize across five very different tasks, but the tasks will serve to show that the choice of conversion scheme has significant impact on down-stream performance.

We used the most recent release of the Mate parser first described in Bohnet (2010),[5] trained on Sections 2–21 of the Wall Street Journal section of the English Treebank (Marcus et al., 1993). The graph-based parser is similar to, except much faster, and performs slightly better than the MSTParser (McDonald et al., 2005), which is known to perform well on long-distance dependencies often important for down-stream applications (McDonald and Nivre, 2007; Galley and Manning, 2009; Bender et al., 2011). This choice may of course have an effect on what conversion schemes seem superior (Johansson and Nugues, 2007). Sentence splitting was done using splitta,[6], and the sentences were then tokenized using PTB-style tokenization[7] and tagged using the in-built Mate POS tagger.

Previous work

There has been considerable work on down-stream evaluation of syntactic parsers in the literature, but most previous work has focused on evaluating parsing models rather than linguistic theories. No one has, to the best of our knowledge, compared the impact of choice of tree-to-dependency conversion scheme across several NLP tasks.

Johansson and Nugues (2007) compare the impact of **yamada** and **lth** on semantic role labeling

[4]The LTH conversion scheme can be obtained by running pennconverter.jar available at http://nlp.cs.lth.se/software/treebank_converter/ with the 'oldLTH' flag set.

[5]http://code.google.com/p/mate-tools/

[6]http://code.google.com/p/splitta/

[7]http://www.cis.upenn.edu/∼treebank/tokenizer.sed

FORM$_1$	FORM$_2$	**yamada**	**conll07**	**ewt**	**lth**
Auxiliary	Main verb	1	1	2	2
Complementizer	Verb	1	2	2	2
Coordinator	Conjuncts	2	1	2	2
Preposition	Nominal	1	1	1	2

Figure 2: Head decisions in conversions. Note: yamada also differ from CoNLL 2007 in proper names.

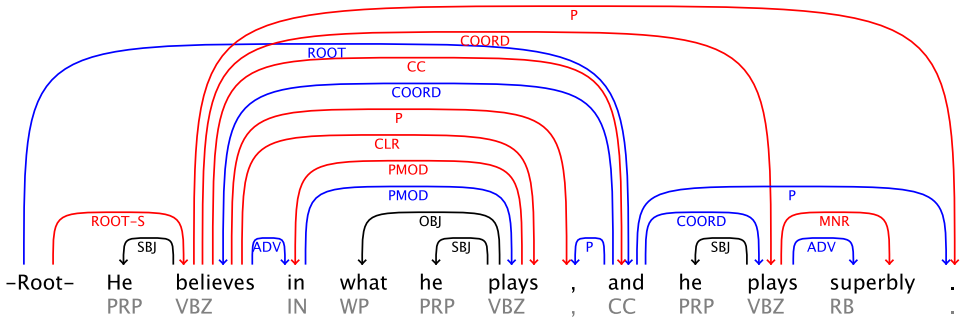

Figure 3: CoNLL 2007 (blue) and LTH (red) dependency conversions.

performance, showing that **lth** leads to superior performance.

Miyao et al. (2008) measure the impact of syntactic parsers in an information extraction system identifying protein-protein interactions in biomedical research articles. They evaluate dependency parsers, constituent-based parsers and deep parsers.

Miwa et al. (2010) evaluate down-stream performance of linguistic representations and parsing models in biomedical event extraction, but do not evaluate linguistic representations directly, evaluating representations and models jointly.

Bender et al. (2011) compare several parsers across linguistic representations on a carefully designed evaluation set of hard, but relatively frequent syntactic constructions. They compare dependency parsers, constituent-based parsers and deep parsers. The authors argue in favor of evaluating parsers on diverse and richly annotated data. Others have discussed various ways of evaluating across annotation guidelines or translating structures to a common format (Schwartz et al., 2011; Tsarfaty et al., 2012).

Hall et al. (2011) discuss optimizing parsers for specific down-stream applications, but consider only a single annotation scheme.

Yuret et al. (2012) present an overview of the SemEval-2010 Evaluation Exercises on Semantic Evaluation track on recognition textual entailment using dependency parsing. They also compare several parsers using the heuristics of the winning system for inference. While the shared task is an example of down-stream evaluation of dependency parsers, the evaluation examples only cover a subset of the textual entailments relevant for practical applications, and the heuristics used in the experiments assume a fixed set of dependency labels (**ewt** labels).

Finally, Schwartz et al. (2012) compare the above conversion schemes and several combinations thereof in terms of learnability. This is very different from what is done here. While learnability may be a theoretically motivated parameter, our results indicate that learnability and downstream performance do not correlate well.

2 Applications

Dependency parsing has proven useful for a wide range of NLP applications, including statistical machine translation (Galley and Manning, 2009; Xu et al., 2009; Elming and Haulrich, 2011) and sentiment analysis (Joshi and Penstein-Rose, 2009; Johansson and Moschitti, 2010). This section describes the applications and experimental set-ups included in this study.

In the five applications considered below we

use syntactic features in slightly different ways. While our statistical machine translation and sentence compression systems use dependency relations as additional information about words and *on a par* with POS, our negation resolution system uses dependency paths, conditioning decisions on both dependency arcs and labels. In perspective classification, we use dependency triples (e.g. SUBJ(John, snore)) as features, while the semantic role labeling system conditions on a lot of information, including the word form of the head, the dependent and the argument candidates, the concatenation of the dependency labels of the predicate, and the labeled dependency relations between predicate and its head, its arguments, dependents or siblings.

2.1 Negation resolution

Negation resolution (NR) is the task of finding negation cues, e.g. the word *not*, and determining their *scope*, i.e. the tokens they affect. NR has recently seen considerable interest in the NLP community (Morante and Sporleder, 2012; Velldal et al., 2012) and was the topic of the 2012 *SEM shared task (Morante and Blanco, 2012).

The data set used in this work, the Conan Doyle corpus (CD),[8] was released in conjunction with the *SEM shared task. The annotations in CD extend on cues and scopes by introducing annotations for in-scope events that are negated in factual contexts. The following is an example from the corpus showing the annotations for cues (bold), scopes (underlined) and negated events (italicized):

(1) Since we have been so
un*fortunate* as to miss him [. . .]

CD-style scopes can be discontinuous and overlapping. Events are a portion of the scope that is semantically negated, with its truth value reversed by the negation cue.

The NR system used in this work (Lapponi et al., 2012), one of the best performing systems in the *SEM shared task, is a CRF model for scope resolution that relies heavily on features extracted from dependency graphs. The feature model contains token distance, direction, n-grams of word forms, lemmas, POS and combinations thereof, as well as the syntactic features presented in Figure 4. The results in our experiments are obtained from configurations that differ only in terms of tree-to-dependency conversions, and are trained on the training set and tested on the development set of CD. Since the negation cue classification component of the system does not rely on dependency features at all, the models are tested using gold cues.

Syntactic	constituent
	dependency relation
	parent head POS
	grand parent head POS
	word form+dependency relation
	POS+dependency relation
Cue-dependent	directed dependency distance
	bidirectional dependency distance
	dependency path
	lexicalized dependency path

Figure 4: Features used to train the conditional random field models

Table 1 shows F_1 scores for scopes, events and full negations, where a true positive correctly assigns both scope tokens and events to the rightful cue. The scores are produced using the evaluation script provided by the *SEM organizers.

2.2 Semantic role labeling

Semantic role labeling (SRL) is the attempt to determine semantic predicates in running text and label their arguments with semantic roles. In our experiments we have reproduced the second best-performing system in the CoNLL 2008 shared task in syntactic and semantic parsing (Johansson and Nugues, 2008).[9]

The English training data for the CoNLL 2008 shared task were obtained from PropBank and NomBank. For licensing reasons, we used OntoNotes 4.0, which includes PropBank, but not NomBank. This means that our system is only trained to classify verbal predicates. We used the Clearparser conversion tool[10] to convert the OntoNotes 4.0 and subsequently supplied syntactic dependency trees using our different conversion schemes. We rely on gold standard argument identification and focus solely on the performance metric semantic labeled F1.

[8] http://www.clips.ua.ac.be/sem2012-st-neg/data.html

[9] http://nlp.cs.lth.se/software/semantic_parsing:_propbank_nombank_frames

[10] http://code.google.com/p/clearparser/

2.3 Statistical machine translation

The effect of the different conversion schemes was also evaluated on SMT. We used the *reordering by parsing* framework described by Elming and Haulrich (2011). This approach integrates a syntactically informed reordering model into a phrase-based SMT system. The model learns to predict the word order of the translation based on source sentence information such as syntactic dependency relations. Syntax-informed SMT is known to be useful for translating between languages with different word orders (Galley and Manning, 2009; Xu et al., 2009), e.g. English and German.

The baseline SMT system is created as described in the guidelines from the original shared task.[11] Only modifications are that we use truecasing instead of lowercasing and recasing, and allow training sentences of up to 80 words. We used data from the English-German restricted task: $\sim$3M parallel words of news, $\sim$46M parallel words of Europarl, and $\sim$309M words of monolingual Europarl and news. We use newstest2008 for tuning, newstest2009 for development, and newstest2010 for testing. Distortion limit was set to 10, which is also where the baseline system performed best. The phrase table and the lexical reordering model is trained on the union of all parallel data with a max phrase length of 7, and the 5-gram language model is trained on the entire monolingual data set.

We test four different experimental systems that only differ with the baseline in the addition of a syntactically informed reordering model. The baseline system was one of the tied best performing system in the WMT 2011 shared task on this dataset. The four experimental systems have reordering models that are trained on the first 25,000 sentences of the parallel news data that have been parsed with each of the tree-to-dependency conversion schemes. The reordering models condition reordering on the word forms, POS, and syntactic dependency relations of the words to be reordered, as described in Elming and Haulrich (2011). The paper shows that while reordering by parsing leads to significant improvements in standard metrics such as BLEU (Papineni et al., 2002) and METEOR (Lavie and Agarwal, 2007), improvements are more spelled out with human judgements. All SMT results reported below are averages based on 5 MERT runs following Clark et al. (2011).

[11] http://www.statmt.org/wmt11/translation-task.html

2.4 Sentence compression

Sentence compression is a restricted form of sentence simplification with numerous usages, including text simplification, summarization and recognizing textual entailment. The most commonly used dataset in the literature is the Ziff-Davis corpus.[12] A widely used baseline for sentence compression experiments is Knight and Marcu (2002), who introduce two models: the noisy-channel model and a decision tree-based model. Both are tree-based methods that find the most likely compressed syntactic tree and outputs the yield of this tree. McDonald et al. (2006) instead use syntactic features to directly find the most likely compressed sentence.

Here we learn a discriminative HMM model (Collins, 2002) of sentence compression using MIRA (Crammer and Singer, 2003), comparable to previously explored models of noun phrase chunking. Our model is thus neither tree-based nor sentence-based. Instead we think of sentence compression as a sequence labeling problem. We compare a model informed by word forms and predicted POS with models also informed by predicted dependency labels. The baseline feature model conditions emission probabilities on word forms and POS using a ± 2 window and combinations thereoff. The augmented syntactic feature model simply adds dependency labels within the same window.

2.5 Perspective classification

Finally, we include a document classification dataset from Lin and Hauptmann (2006).[13] The dataset consists of blog posts posted at bitterlemons.org by Israelis and Palestinians. The bitterlemons.org website is set up to "contribute to mutual understanding through the open exchange of ideas." In the dataset, each blog post is labeled as either Israeli or Palestinian. Our baseline model is just a standard bag-of-words model, and the system adds dependency triplets to the bag-of-words model in a way similar to Joshi and Penstein-Rose (2009). We do not remove stop words, since perspective classification is

[12] LDC Catalog No.: LDC93T3A.

[13] https://sites.google.com/site/weihaolinatcmu/data

	bl	**yamada**	**conll07**	**ewt**	**lth**
DEPRELS	-	12	21	47	41
PTB-23 (LAS)	-	**88.99**	88.52	81.36*	87.52
PTB-23 (UAS)	-	90.21	90.12	84.22*	**90.29**
Neg: scope F_1	-	**81.27**	80.43	78.70	79.57
Neg: event F_1	-	76.19	72.90	73.15	**76.24**
Neg: full negation F_1	-	**67.94**	63.24	61.60	64.31
SentComp F_1	68.47	**72.07**	64.29	71.56	71.56
SMT-dev-Meteor	35.80	36.06	36.06	**36.16**	36.08
SMT-test-Meteor	37.25	37.48	37.50	**37.58**	37.51
SMT-dev-BLEU	13.66	**14.14**	14.09	14.04	14.06
SMT-test-BLEU	14.67	15.04	15.04	14.96	**15.11**
SRL-22-gold	-	81.35	83.22	**84.72**	84.01
SRL-23-gold	-	79.09	80.85	80.39	**82.01**
SRL-22-pred	-	74.41	76.22	**78.29**	66.32
SRL-23-pred	-	73.42	74.34	**75.80**	64.06
bitterlemons.org	96.08	**97.06**	95.58	96.08	96.57

Table 1: Results. *: Low parsing results on PTB-23 using **ewt** are explained by changes between the PTB-III and the Ontonotes 4.0 release of the English Treebank.

similar to authorship attribution, where stop words are known to be informative. We evaluate performance doing cross-validation over the official training data, setting the parameters of our learning algorithm for each fold doing cross-validation over the actual training data. We used soft-margin support vector machine learning (Cortes and Vapnik, 1995), tuning the kernel (linear or polynomial with degree 3) and $C = \{0.1, 1, 5, 10\}$.

3 Results and discussion

Our results are presented in Table 1. The parsing results are obtained relying on predicted POS rather than, as often done in the dependency parsing literature, relying on gold-standard POS. Note that they comply with the result in Schwartz et al. (2012) that Yamada-Matsumoto-style annotation is more easily learnable.

The **negation resolution** results are significantly better using syntactic features in **yamada** annotation. It is not surprising that a syntactically oriented conversion scheme performs well in this task. Since Lapponi et al. (2012) used Maltparser (Nivre et al., 2007) with the freely available pre-trained parsing model for English,[14] we decided to also run that parser with the gold-standard cues, in addition to Mate. The pre-trained model was trained on Sections 2–21 of the Wall Street Journal section of the English Treebank (Marcus et al., 1993), augmented with 4000 sentences from the QuestionBank,[15] which was converted using the Stanford converter and thus similar to the **ewt** annotations used here. The results were better than using **ewt** with Mate trained on Sections 2–21 alone, but worse than the results obtained here with **yamada** conversion scheme. F_1 score on full negation was 66.92%.

The case-sensitive BLEU evaluation of the **SMT** systems indicates that choice of conversion scheme has no significant impact on overall performance. The difference to the baseline system is significant ($p < 0.01$), showing that the reordering model leads to improvement using any of the schemes. However, the conversion schemes lead to very different translations. This can be seen, for example, by the fact that the relative tree edit distance between translations of different syntactically informed SMT systems is 12% higher than within each system (across different MERT optimizations).

The reordering approach puts a lot of weight on the syntactic dependency relations. As a consequence, the number of relation types used in the conversion schemes proves important. Consider the

[14] http://www.maltparser.org/mco/english_parser/engmalt.html

[15] http://www.computing.dcu.ie/∼jjudge/qtreebank/

REFERENCE:	Zum Glück kam ich beim Strassenbahnfahren an die richtige Stelle .
SOURCE:	Luckily , on the way to the tram , I found the right place .
yamada:	Glücklicherweise hat auf dem Weg zur S-Bahn , stellte ich fest , dass der richtige Ort .
conll07:	Glücklicherweise hat auf dem Weg zur S-Bahn , stellte ich fest , dass der richtige Ort .
ewt:	Zum Glück fand ich auf dem Weg zur S-Bahn , am richtigen Platz .
lth:	Zum Glück fand ich auf dem Weg zur S-Bahn , am richtigen Platz .
BASELINE:	Zum Glück hat auf dem Weg zur S-Bahn , ich fand den richtigen Platz .

Figure 5: Examples of SMT output.

ORIGINAL:	* 68000 sweden ab of uppsala , sweden , introduced the teleserve , an integrated answering machine and voice-message handler that links a macintosh to touch-tone phones .
BASELINE:	68000 sweden ab introduced the teleserve an integrated answering machine and voice-message handler .
yamada	68000 sweden ab introduced the teleserve integrated answering machine and voice-message handler .
conll07	68000 sweden ab **sweden** introduced the teleserve integrated answering machine and voice-message handler .
ewt	68000 sweden ab introduced the teleserve integrated answering machine and voice-message handler .
lth	68000 sweden ab introduced the teleserve **an** integrated answering machine and voice-message handler .
HUMAN:	68000 sweden ab introduced the teleserve integrated answering machine and voice-message handler .

Figure 6: Examples of sentence compression output.

example in Figure 5. German requires the verb in second position, which is obeyed in the much better translations produced by the **ewt** and **lth** systems. Interestingly, the four schemes produce virtually identical structures for the source sentence, but they differ in their labeling. Where **conll07** and **yamada** use the same relation for the first two constituents (ADV and vMOD, respectively), **ewt** and **lth** distinguish between them (ADVMOD/PREP and ADV/LOC). This distinction may be what enables the better translation, since the model may learn to move the verb after the sentence adverbial. In the other schemes, sentence adverbials are not distinguished from locational adverbials. Generally, **ewt** and **lth** have more than twice as many relation types as the other schemes.

The schemes **ewt** and **lth** lead to better **SRL** performance than **conll07** and **yamada** when relying on gold-standard syntactic dependency trees. This supports the claims put forward in Johansson and Nugues (2007). These annotations also happen to use a larger set of dependency labels, however, and syntactic structures may be harder to reconstruct, as reflected by labeled attachment scores (LAS) in syntactic parsing. The biggest drop in SRL performance going from gold-standard to predicted syntactic trees is clearly for the **lth** scheme, at an average 17.8% absolute loss (**yamada** 5.8%; **conll07** 6.8%; **ewt** 5.5%; **lth** 17.8%).

The **ewt** scheme resembles **lth** in most respects, but in preposition-noun dependencies it marks the preposition as the head rather than the noun. This is an important difference for SRL, because semantic arguments are often nouns embedded in prepositional phrases, like agents in passive constructions. It may also be that the difference in performance is simply explained by the syntactic analysis of prepositional phrases being easier to reconstruct.

The **sentence compression** results are generally much better than the models proposed in Knight and Marcu (2002). Their noisy channel model obtains an F_1 compression score of 14.58%, whereas the decision tree-based model obtains an F_1 compression score of 31.71%. While F_1 scores should be complemented by human judgements, as there are typically many good sentence compressions of any source sentence, we believe that error reductions of more than 50% indicate that the models used here

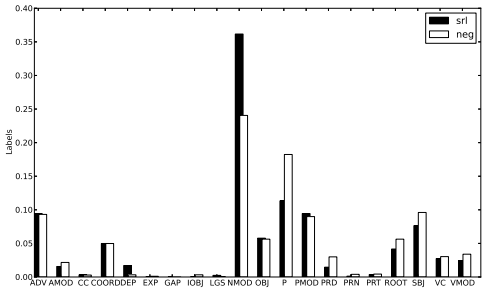

Figure 7: Distributions of dependency labels in the Yamada-Matsumoto scheme

(though previously unexplored in the literature) are fully competitive with state-of-the-art models.

We also see that the models using syntactic features perform better than our baseline model, except for the model using **conll07** dependency annotation. This may be surprising to some, since distributional information is often considered important in sentence compression (Knight and Marcu, 2002). Some output examples are presented in Figure 6. Unsurprisingly, it is seen that the baseline model produces grammatically incorrect output, and that most of our syntactic models correct the error leading to ungrammaticality. The model using **ewt** annotation is an exception. We also see that **conll07** introduces another error. We believe that this is due to the way the **conll07** tree-to-dependency conversion scheme handles coordination. While the word *Sweden* is not coordinated, it occurs in a context, surrounded by commas, that is very similar to coordinated items.

In **perspective classification** we see that syntactic features based on **yamada** and **lth** annotations lead to improvements, with **yamada** leading to slightly better results than **lth**. The fact that a syntactically oriented conversion scheme leads to the best results may reflect that perspective classification, like authorship attribution, is less about content than stylistics.

While **lth** seems to lead to the overall best results, we stress the fact that the five tasks considered here are incommensurable. What is more interesting is that, task to task, results are so different. The semantically oriented conversion schemes, **ewt** and **lth**, lead to the best results in SRL, but with a significant drop for **lth** when relying on predicted parses, while the **yamada** scheme is competitive in the other four tasks. This may be because distributional information is more important in these tasks than in SRL.

The distribution of dependency labels seems relatively stable across applications, but differences in data may of course also affect the usefulness of different annotations. Note that **conll07** leads to very good results for negation resolution, but bad results for SRL. See Figure 7 for the distribution of labels in the **conll07** conversion scheme on the SRL and negation scope resolution data. Many differences relate to differences in sentence length. The negation resolution data is literary text with shorter sentences, which therefore uses more punctuation and has more root dependencies than newspaper articles. On the other hand we do see very few predicate dependencies in the SRL data. This may affect down-stream results when classifying verbal predicates in SRL. We also note that the number of dependency labels have less impact on results in general than we would have expected. The number of dependency labels and the lack of support for some of them may explain the drop with predicted syntactic parses in our SRL results, but generally we obtain our best results with **yamada** and **lth** annotations, which have 12 and 41 dependency labels, respectively.

4 Conclusions

We evaluated four different tree-to-dependency conversion schemes, putting more or less emphasis on syntactic or semantic evidence, in five down-stream applications, including SMT and negation resolution. Our results show why it is important to be precise about exactly what tree-to-dependency conversion scheme is used. Tools like pennconverter.jar gives us a wide range of options when converting constituent-based treebanks, and even small differences may have significant impact on down-stream performance. The small differences are also important for more linguistic comparisons that also tend to gloss over exactly what conversion scheme is used, e.g. Ivanova et al. (2012).

Acknowledgements

Hector Martinez is funded by the ERC grant CLARA No. 238405, and Anders Søgaard is funded by the ERC Starting Grant LOWLANDS No. 313695.

References

Emily Bender, Dan Flickinger, Stephan Oepen, and Yi Zhang. 2011. Parser evaluation over local and non-local dependencies in a large corpus. In *EMNLP*.

Bernd Bohnet. 2010. Top accuracy and fast dependency parsing is not a contradiction. In *COLING*.

Jonathan H. Clark, Chris Dyer, Alon Lavie, and Noah A. Smith. 2011. Better hypothesis testing for statistical machine translation: controlling for optimizer instability. In *ACL*.

Mike Collins. 1999. *Head-driven statistical models for natural language parsing*. Ph.D. thesis, University of Pennsylvania.

Michael Collins. 2002. Discriminative training methods for Hidden Markov Models. In *EMNLP*.

Corinna Cortes and Vladimir Vapnik. 1995. Support-vector networks. *Machine Learning*, 20(3):273–297.

Koby Crammer and Yoram Singer. 2003. Ultraconservative algorithms for multiclass problems. In *JMLR*.

Jakob Elming and Martin Haulrich. 2011. Reordering by parsing. In *Proceedings of International Workshop on Using Linguistic Information for Hybrid Machine Translation (LIHMT-2011)*.

Michel Galley and Christopher Manning. 2009. Quadratic-time dependency parsing for machine translation. In *ACL*.

Keith Hall, Ryan McDonald, Jason Katz-Brown, and Michael Ringgaard. 2011. Training dependency parsers by jointly optimizing multiple objectives. In *EMNLP*.

Angelina Ivanova, Stephan Oepen, Lilja Øvrelid, and Dan Flickinger. 2012. Who did what to whom? a contrastive study of syntactico-semantic dependencies. In *LAW*.

Richard Johansson and Alessandro Moschitti. 2010. Syntactic and semantic structure for opinion expression detection. In *CoNLL*.

Richard Johansson and Pierre Nugues. 2007. Extended constituent-to-dependency conversion for English. In *NODALIDA*.

Richard Johansson and Pierre Nugues. 2008. Dependency-based syntactic-semantic analysis with propbank and nombank. In *CoNLL*.

Mahesh Joshi and Carolyn Penstein-Rose. 2009. Generalizing dependency features for opinion mining. In *ACL*.

Kevin Knight and Daniel Marcu. 2002. Summarization beyond sentence extraction: a probabilistic approach to sentence compression. *Artificial Intelligence*, 139:91–107.

Emanuele Lapponi, Erik Velldal, Lilja Øvrelid, and Jonathon Read. 2012. UiO2: Sequence-labeling negation using dependency features. In **SEM*.

Alon Lavie and Abhaya Agarwal. 2007. Meteor: an automatic metric for mt evaluation with high levels of correlation with human judgments. In *WMT*.

Wei-Hao Lin and Alexander Hauptmann. 2006. Are these documents written from different perspectives? In *COLING-ACL*.

Mitchell Marcus, Mary Marcinkiewicz, and Beatrice Santorini. 1993. Building a large annotated corpus of English: the Penn Treebank. *Computational Linguistics*, 19(2):313–330.

Ryan McDonald and Joakim Nivre. 2007. Characterizing the errors of data-driven dependency parsers. In *EMNLP-CoNLL*.

Ryan McDonald, Fernando Pereira, Kiril Ribarov, and Jan Hajič. 2005. Non-projective dependency parsing using spanning tree algorithms. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing 2005*, pages 523–530, Vancouver, British Columbia.

Ryan McDonald. 2006. Discriminative sentence compression with soft syntactic evidence. In *EACL*.

Makoto Miwa, Sampo Pyysalo, Tadayoshi Hara, and Jun'ichi Tsujii. 2010. Evaluating dependency representation for event extraction. In *COLING*.

Yusuke Miyao, Rune Sæ tre, Kenji Sagae, Takuya Matsuzaki, and Jun'ichi Tsujii. 2008. Task-oriented evaluation of syntactic parsers and their representations. In *ACL*.

Roser Morante and Eduardo Blanco. 2012. *sem 2012 shared task: Resolving the scope and focus of negation. In **SEM*.

Roser Morante and Caroline Sporleder. 2012. Modality and negation: An introduction to the special issue. *Computational linguistics*, 38(2):223–260.

Joakim Nivre, Johan Hall, Jens Nilsson, 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(2):95–135.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *ACL*.

Slav Petrov and Ryan McDonald. 2012. Overview of the 2012 Shared Task on Parsing the Web. In *Notes of the First Workshop on Syntactic Analysis of Non-Canonical Language (SANCL)*.

Roy Schwartz, and Omri Abend, Roi Reichart, and Ari Rappoport. 2011. Neutralizing linguistically problematic annotations in unsupervised dependency parsing evaluation. In *ACL*.

Roy Schwartz, Omri Abend, and Ari Rappoport. 2012. Learnability-based syntactic annotation design. In *COLING*.

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

Erik Velldal, Lilja Øvrelid, Jonathon Read, and Stephan Oepen. 2012. Speculation and negation: Rules, rankers, and the role of synta. *Computational linguistics*, 38(2):369–410.

Peng Xu, Jaeho Kang, Michael Ringgaard, and Franz Och. 2009. Using a dependency parser to improve SMT for subject-object-verb languages. In *NAACL-HLT*, Boulder, Colorado.

Deniz Yuret, Laura Rimell, and Aydin Han. 2012. Parser evaluation using textual entailments. *Language Resources and Evaluation*, Published online 31 October 2012.