A Comparative Study of Target Dependency Structures for Statistical Machine Translation

Xianchao Wu,[*] Katsuhito Sudoh, Kevin Duh,[†] Hajime Tsukada, Masaaki Nagata
NTT Communication Science Laboratories, NTT Corporation
2-4 Hikaridai Seika-cho, Soraku-gun Kyoto 619-0237 Japan
wuxianchao@gmail.com,sudoh.katsuhito@lab.ntt.co.jp,
kevinduh@is.naist.jp,{tsukada.hajime,nagata.masaaki}@lab.ntt.co.jp

Abstract

This paper presents a comparative study of target dependency structures yielded by several state-of-the-art linguistic parsers. Our approach is to measure the impact of these non-isomorphic dependency structures to be used for string-to-dependency translation. Besides using traditional dependency parsers, we also use the dependency structures transformed from PCFG trees and predicate-argument structures (PASs) which are generated by an HPSG parser and a CCG parser. The experiments on Chinese-to-English translation show that the HPSG parser's PASs achieved the best dependency and translation accuracies.

1 Introduction

Target language side dependency structures have been successfully used in statistical machine translation (SMT) by Shen et al. (2008) and achieved state-of-the-art results as reported in the NIST 2008 Open MT Evaluation workshop and the NTCIR-9 Chinese-to-English patent translation task (Goto et al., 2011; Ma and Matsoukas, 2011). A primary advantage of dependency representations is that they have a natural mechanism for representing discontinuous constructions, which arise due to long-distance dependencies or in languages where grammatical relations are often signaled by morphology instead of word order (McDonald and Nivre, 2011).

It is known that dependency-style structures can be transformed from a number of linguistic structures. For example, using the constituent-to-dependency conversion approach proposed by Johansson and Nugues (2007), we can easily yield dependency trees from PCFG style trees. A semantic dependency representation of a whole sentence, predicate-argument structures (PASs), are also included in the output trees of (1) a state-of-the-art head-driven phrase structure grammar (HPSG) (Pollard and Sag, 1994; Sag et al., 2003) parser, Enju[1] (Miyao and Tsujii, 2008) and (2) a state-of-the-art CCG parser[2] (Clark and Curran, 2007). The motivation of this paper is to investigate the impact of these non-isomorphic dependency structures to be used for SMT. That is, we would like to provide a comparative evaluation of these dependencies in a string-to-dependency decoder (Shen et al., 2008).

2 Gaining Dependency Structures

2.1 Dependency tree

We follow the definition of *dependency graph* and *dependency tree* as given in (McDonald and Nivre, 2011). A dependency graph G for sentence s is called a *dependency tree* when it satisfies, (1) the nodes cover all the words in s besides the ROOT; (2) one node can have one and only one head (word) with a determined syntactic role; and (3) the ROOT of the graph is reachable from all other nodes.

For extracting string-to-dependency transfer rules, we use *well-formed* dependency structures, either fixed or floating, as defined in (Shen et al., 2008). Similarly, we ignore the syntactic roles

[*] Now at Baidu Inc.
[†] Now at Nara Institute of Science & Technology (NAIST)

[1] http://www-tsujii.is.s.u-tokyo.ac.jp/enju/index.html
[2] http://groups.inf.ed.ac.uk/ccg/software.html

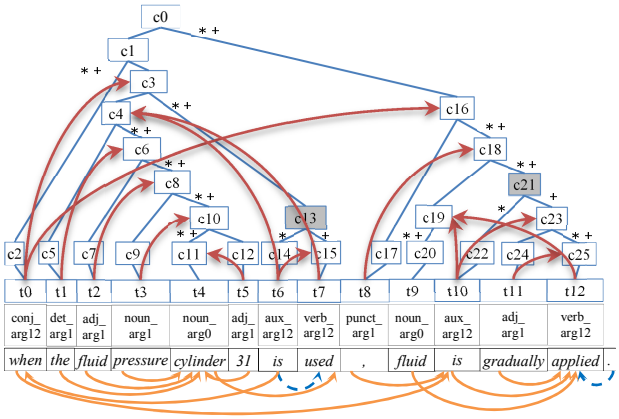

Figure 1: HPSG tree of an example sentence. '*'/ '+'=syntactic/semantic heads. Arrows in red (upper)= PASs, orange (bottom)=word-level dependencies generated from PASs, blue=newly appended dependencies.

both during rule extracting and target dependency language model (LM) training.

2.2 Dependency parsing

Graph-based and transition-based are two predominant paradigms for data-driven dependency parsing. The MST parser (McDonald et al., 2005) and the Malt parser (Nivre, 2003) stand for two typical parsers, respectively. Parsing accuracy comparison and error analysis under the CoNLL-X dependency shared task data (Buchholz and Marsi, 2006) have been performed by McDonald and Nivre (2011). Here, we compare them on the SMT tasks through parsing the real-world SMT data.

2.3 PCFG parsing

For PCFG parsing, we select the Berkeley parser (Petrov and Klein, 2007). In order to generate word-level dependency trees from the PCFG tree, we use the LTH constituent-to-dependency conversion tool[3] written by Johansson and Nugues (2007). The head finding rules[4] are according to Magerman (1995) and Collins (1997). Similar approach has been originally used by Shen et al. (2008).

2.4 HPSG parsing

In the Enju English HPSG grammar (Miyao et al., 2003) used in this paper, the semantic content of a sentence/phrase is represented by a PAS. In an HPSG tree, each leaf node generally introduces a predicate, which is represented by the pair made up of the lexical entry feature and predicate type feature. The arguments of a predicate are designated by the arrows from the argument features in a leaf node to non-terminal nodes (e.g., t0→c3, t0→c16).

Since the PASs use the non-terminal nodes in the HPSG tree (Figure 1), this prevents their direct usage in a string-to-dependency decoder. We thus need an algorithm to transform these phrasal predicate-argument dependencies into a word-to-word dependency tree. Our algorithm (refer to Figure 1 for an example) for changing PASs into word-based dependency trees is as follows:

1. *finding*, i.e., find the syntactic/semantic head word of each argument node through a bottom-up traversal of the tree;

2. *mapping*, i.e., determine the *arc directions* (among a predicate word and the syntactic/semantic head words of the argument nodes) for each predicate type according to Table 1. Then, a dependency graph will be generated;

3. *checking*, i.e., post modifying the dependency graph according to the definition of *dependency tree* (Section 2.1).

Table 1 lists the mapping from HPSG's PAS types to word-level dependency arcs. Since a non-terminal node in an HPSG tree has two kinds of heads, syntactic or semantic, we will generate two dependency graphs after mapping. We use "PAS+syn" to represent the dependency trees generated from the HPSG PASs guided by the syntactic heads. For semantic heads, we use "PAS+sem".

For example, refer to t0 = when in Figure 1. Its arg1 = c16 (with syntactic head t10), arg2 = c3 (with syntactic head t6), and PAS type = conj_arg12. In Table 1, this PAS type corresponds to arg2→pred→arg1, then the result word-level dependency is t6(is)→t0(when)→t10(is).

We need to post modify the dependency graph after applying the mapping, since it is not guaranteed to be a dependency tree. Referring to the definition of dependency tree (Section 2.1), we need the strategy for (1) selecting only one head from multiple

[3] http://nlp.cs.lth.se/software/treebank_converter/

[4] http://www.cs.columbia.edu/ mcollins/papers/heads

PAS Type	Dependency Relation
adj_arg1[2]	[arg2 →] pred → arg1
adj_mod_arg1[2]	[arg2 →] pred → arg1 → mod
aux[_mod]_arg12	arg1/pred → arg2 [→ mod]
conj_arg1[2[3]]	[arg2[/arg3]] → pred → arg1
comp_arg1[2]	pred → arg1 [→ arg2]
comp_mod_arg1	arg1 → pred → mod
noun_arg1	pred → arg1
noun_arg[1]2	arg2 → pred [→ arg1]
poss_arg[1]2	pred → arg2 [→ arg1]
prep_arg12[3]	arg2[/arg3] → pred → arg1
prep_mod_arg12[3]	arg2[/arg3] → pred → arg1 → mod
quote_arg[1]2	[arg1 →] pred → arg2
quote_arg[1]23	[arg1/]arg3 → pred → arg2
lparen_arg123	pred/arg2 → arg3 → arg1
relative_arg1[2]	[arg2 →] pred → arg1
verb_arg1[2[3[4]]]	arg1[/arg2[/arg3[/arg4]]] → pred
verb_mod_arg1[2[3[4]]]	arg1[/arg2[/arg3[/arg4]]]→pred→mod
app_arg12,coord_arg12	arg2/pred → arg1
det_arg1,it_arg1,punct_arg1	pred → arg1
dtv_arg2	pred → arg2
lgs_arg2	arg2 → pred

Table 1: Mapping from HPSG's PAS types to dependency relations. Dependent(s) → head(s), / = and, [] = optional.

heads and (2) appending dependency relations for those words/punctuation that do not have any head. When one word has multiple heads, we only keep one. The selection strategy is that, if this arc was deleted, it will cause the biggest number of words that can not reach to the root word anymore. In case of a tie, we greedily pack the arc that connect two words w_i and w_j where $|i-j|$ is the biggest. For all the words and punctuation that do not have a head, we greedily take the root word of the sentence as their heads. In order to fully use the training data, if there are directed cycles in the result dependency graph, we still use the graph in our experiments, where only partial dependency arcs, i.e., those target flat/hierarchical phrases attached with well-formed dependency structures, can be used during translation rule extraction.

2.5 CCG parsing

We also use the predicate-argument dependencies generated by the CCG parser developed by Clark and Curran (2007). The algorithm for generating word-level dependency tree is easier than processing the PASs included in the HPSG trees, since the word level predicate-argument relations have already been included in the output of CCG parser. The mapping from predicate types to the gold-standard grammatical relations can be found in Table 13 in (Clark and Curran, 2007). The post-processing is like that described for HPSG parsing, except we greedily use the MST's sentence root when we can not determine it based on the CCG parser's PASs.

3 Experiments

3.1 Setup

We re-implemented the string-to-dependency decoder described in (Shen et al., 2008). Dependency structures from non-isomorphic syntactic/semantic parsers are separately used to train the transfer rules as well as target dependency LMs. For intuitive comparison, an outside SMT system is Moses (Koehn et al., 2007).

For Chinese-to-English translation, we use the parallel data from NIST Open Machine Translation Evaluation tasks. The training data contains 353,796 sentence pairs, 8.7M Chinese words and 10.4M English words. The NIST 2003 and 2005 test data are respectively taken as the development and test set. We performed GIZA++ (Och and Ney, 2003) and the *grow-diag-final-and* symmetrizing strategy (Koehn et al., 2007) to obtain word alignments. The Berkeley Language Modeling Toolkit, berkeleylm-1.0b3[5] (Pauls and Klein, 2011), was employed to train (1) a five-gram LM on the Xinhua portion of LDC English Gigaword corpus v3 (LDC2007T07) and (2) a tri-gram dependency LM on the English dependency structures of the training data. We report the translation quality using the case-insensitive BLEU-4 metric (Papineni et al., 2002).

3.2 Statistics of dependencies

We compare the similarity of the dependencies with each other, as shown in Table 2. Basically, we investigate (1) if two dependency graphs of one sentence share the same root word and (2) if the head of one word in one sentence are identical in two dependency graphs. In terms of root word comparison, we observe that MST and CCG share 87.3% of identical root words, caused by borrowing roots from MST to CCG. Then, it is interesting that Berkeley and PAS+syn share 74.8% of identical root words. Note that the Berkeley parser is trained on the Penn treebank (Marcus et al., 1994) yet the HPSG parser is trained on the HPSG treebank (Miyao and Tsujii,

[5]http://code.google.com/p/berkeleylm/

Dependency	Precision	Recall	BLEU-Dev	BLEU-Test	# phrases	# hier rules	# illegal dep trees	# directed cycles
Moses-1	-	-	0.3349	0.3207	5.4M	-	-	-
Moses-2	-	-	0.3445	0.3262	0.7M	4.5M	-	-
MST	0.744	0.750	**0.3520**	0.3291	2.4M	2.1M	251	0
Malt	0.732	0.738	0.3423	0.3203	1.5M	1.3M	130,960	0
Berkeley	0.800	0.806	0.3475	0.3312	2.4M	2.2M	282	0
PAS+syn	**0.818**	**0.824**	0.3499	**0.3376**	2.2M	1.9M	10,411	5,853
PAS+sem	0.777	0.782	0.3484	0.3343	2.1M	1.6M	14,271	9,747
CCG	0.701	0.705	0.3442	0.3283	1.7M	1.3M	61,015	49,955

Table 3: Comparison of dependency and translation accuracies. Moses-1 = phrasal, Moses-2 = hierarchical.

	Malt	Berkeley	PAS +syn	PAS +sem	CCG
MST	70.5 (**77.3**)	62.5 (64.6)	69.2 (58.5)	53.3 (58.1)	**87.3** (61.7)
Malt		66.2 (63.2)	73.0 (57.7)	46.8 (56.6)	62.9 (58.1)
Berkeley			**74.8** (64.3)	44.2 (56.0)	56.5 (59.2)
PAS+ syn				59.3 (**79.1**)	62.9 (61.0)
PAS+ sem					60.0 (58.8)

Table 2: Comparison of the dependencies of the English sentences in the training data. Without () = % of similar root words; with () = % of similar head words.

2008). In terms of head word comparison, PAS+syn and PAS+sem share 79.1% of identical head words. This is basically due to that we used the similar PASs of the HPSG trees. Interestingly, there are only 59.3% identical root words shared by PAS+syn and PAS+sem. This reflects the significant difference between syntactic and semantic heads.

We also manually created the golden dependency trees for the first 200 English sentences in the training data. The precision/recall (P/R) are shown in Table 3. We observe that (1) the translation accuracies approximately follow the P/R scores yet are not that sensitive to their large variances, and (2) it is still tough for domain-adapting from the treebank-trained parsers to parse the real-world SMT data. PAS+syn performed the best by avoiding the errors of missing of arguments for a predicate, wrongly identified head words for a linguistic phrase, and inconsistency dependencies inside relatively long coordinate structures. These errors significantly influence the number of extractable translation rules and the final translation accuracies.

Note that, these P/R scores on the first 200 sentences (all from less than 20 newswire documents) shall only be taken as an approximation of the total training data and not necessarily exactly follow the tendency of the final BLEU scores. For example, CCG is worse than Malt in terms of P/R yet with a higher BLEU score. We argue this is mainly due to that the number of illegal dependency trees generated by Malt is the highest. Consequently, the number of flat/hierarchical rules generated by using Malt trees is the lowest. Also, PAS+sem has a lower P/R than Berkeley, yet their final BLEU scores are not statistically different.

3.3 Results

Table 3 also shows the BLEU scores, the number of flat phrases and hierarchical rules (both integrated with target dependency structures), and the number of illegal dependency trees generated by each parser. From the table, we have the following observations: (1) all the dependency structures (except Malt) achieved a significant better BLEU score than the phrasal Moses; (2) PAS+syn performed the best in the test set (0.3376), and it is significantly better than phrasal/hierarchical Moses ($p < 0.01$), MST ($p < 0.05$), Malt ($p < 0.01$), Berkeley ($p < 0.05$), and CCG ($p < 0.05$); and (3) CCG performed as well as MST and Berkeley. These results lead us to argue that the robustness of deep syntactic parsers can be advantageous in SMT compared with traditional dependency parsers.

4 Conclusion

We have constructed a string-to-dependency translation platform for comparing non-isomorphic target dependency structures. Specially, we proposed an algorithm for generating word-based dependency trees from PASs which are generated by a state-of-the-art HPSG parser. We found that dependency trees transformed from these HPSG PASs achieved the best dependency/translation accuracies.

References

Sabine Buchholz and Erwin Marsi. 2006. Conll-x shared task on multilingual dependency parsing. In *Proceedings of the Tenth Conference on Computational Natural Language Learning (CoNLL-X)*, pages 149–164, New York City, June. Association for Computational Linguistics.

Stephen Clark and James R. Curran. 2007. Wide-coverage efficient statistical parsing with ccg and log-linear models. *Computational Linguistics*, 33(4):493–552.

Michael Collins. 1997. Three generative, lexicalised models for statistical parsing. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics*, pages 16–23, Madrid, Spain, July. Association for Computational Linguistics.

Isao Goto, Bin Lu, Ka Po Chow, Eiichiro Sumita, and Benjamin K. Tsou. 2011. Overview of the patent machine translation task at the ntcir-9 workshop. In *Proceedings of NTCIR-9*, pages 559–578.

Richard Johansson and Pierre Nugues. 2007. Extended constituent-to-dependency conversion for english. In *In Proceedings of NODALIDA*, Tartu, Estonia, April.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the ACL 2007 Demo and Poster Sessions*, pages 177–180.

Jeff Ma and Spyros Matsoukas. 2011. Bbn's systems for the chinese-english sub-task of the ntcir-9 patentmt evaluation. In *Proceedings of NTCIR-9*, pages 579–584.

David Magerman. 1995. Statistical decision-tree models for parsing. In *In Proceedings of of the 33rd Annual Meeting of the Association for Computational Linguistics*, pages 276–283.

Mitchell Marcus, Grace Kim, Mary Ann Marcinkiewicz, Robert MacIntyre, Ann Bies, Mark Ferguson, Karen Katz, and Britta Schasberger. 1994. The penn treebank: Annotating predicate argument structure. In *Proceedings of the Workshop on HLT*, pages 114–119, Plainsboro.

Ryan McDonald and Joakim Nivre. 2011. Analyzing and integrating dependency parsers. *Computational Linguistics*, 37(1):197–230.

Ryan McDonald, Koby Crammer, and Fernando Pereira. 2005. Online large-margin training of dependency parsers. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 91–98, Ann Arbor, Michigan, June. Association for Computational Linguistics.

Yusuke Miyao and Jun'ichi Tsujii. 2008. Feature forest models for probabilistic hpsg parsing. *Computational Lingustics*, 34(1):35–80.

Yusuke Miyao, Takashi Ninomiya, and Jun'ichi Tsujii. 2003. Probabilistic modeling of argument structures including non-local dependencies. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing*, pages 285–291, Borovets.

Joakim Nivre. 2003. An efficient algorithm for projective dependency parsing. In *Proceedings of the 8th International Workshop on Parsing Technologies (IWPT*, pages 149–160.

Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of ACL*, pages 311–318.

Adam Pauls and Dan Klein. 2011. Faster and smaller n-gram language models. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 258–267, Portland, Oregon, USA, June. Association for Computational Linguistics.

Slav Petrov and Dan Klein. 2007. Improved inference for unlexicalized parsing. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, pages 404–411, Rochester, New York, April. Association for Computational Linguistics.

Carl Pollard and Ivan A. Sag. 1994. *Head-Driven Phrase Structure Grammar*. University of Chicago Press.

Ivan A. Sag, Thomas Wasow, and Emily M. Bender. 2003. *Syntactic Theory: A Formal Introduction*. Number 152 in CSLI Lecture Notes. CSLI Publications.

Libin Shen, Jinxi Xu, and Ralph Weischedel. 2008. A new string-to-dependency machine translation algorithm with a target dependency language model. In *Proceedings of ACL-08:HLT*, pages 577–585, Columbus, Ohio.