Initial Explorations of CCG Supertagging for Universal Dependency Parsing

Burak Kerim Akkus, Heval Azizoglu and Ruket Cakici
Computer Engineering Department
Middle East Technical University
Ankara, Turkey
{burakkerim,heval.azizoglu,ruken}@ceng.metu.edu.tr

Abstract

In this paper we describe the system by METU team for universal dependency parsing of multilingual text. We use a neural network-based dependency parser that has a greedy transition approach to dependency parsing. CCG supertags contain rich structural information that proves useful in certain NLP tasks. We experiment with CCG supertags as additional features in our experiments. The neural network parser is trained together with dependencies and simplified CCG tags as well as other features provided.

1 Introduction

Combinatory Categorial Grammar (Steedman, 2000) (CCG) is widely used for natural language processing for its desirable properties of generative expressiveness and its transparent interface of syntax and underlying semantic interpretation. CCG has been used for creating fast and accurate parsers (Hockenmaier and Steedman, 2002), (Clark and Curran, 2007), (Auli and Lopez, 2011), (Lewis and Steedman, 2014). In addition to this, the structural information in the CCG categories, which is a lexicalised grammar, has been shown to improve performance of various other systems when used indirectly. Examples are multilingual dependency parsing and machine translation (Ambati et al., 2013a; Çakıcı, 2008; Birch et al., 2007).

In this paper, we describe a system we created for CoNLL Shared Task of 2017 (Zeman et al., 2017) Multilingual Parsing from Raw Text to Universal Dependencies (Nivre et al., 2016, 2017b) [1] . We use CCG categories induced from the CCGBank (Hockenmaier and Steedman, 2007) to supertag different languages with these structural information-packed tags. We aim to show that CCG categories for English may be used to improve parsing results for other languages, especially similar ones.

In the next section, we give a brief background on the dependency parsing problem and CCG categories that have been shown to improve performance on various tasks either directly or indirectly. Section 3 gives the implementation details for the METU system and in Section 4 results are discussed.

2 Background

Combinatory Categorial Grammar is a lexicalised grammar formalism that has a transparent syntax-semantics interface which means one can create rich semantic interpretations in parallel with parsing (Steedman, 2000). Several fast and highly accurate CCG parsers have been introduced in the literature. These parsers make use of the CCG-Bank (Hockenmaier and Steedman, 2002) that is created by inducing a CCG Grammar from the Penn Treebank (Marcus et al., 1993). The CCG categories of each word are extracted and referred to as supertags or CCG tags throughout the paper.

Different types of supertags such as the ones encoding predicate argument structure or morphosyntactic information have been shown to increase parsing performance in several studies starting with Bangalore and Joshi (1999). The importance of supertagging in parsing accuracy has been shown in various studies such as Falenska et al. (2015), Ouchi et al. (2014) and Foth et al. (2006) for different types of supertags such as combinations of dependency labels and dependent positions, and by Clark and Curran (2004) for CCG categories as supertags. The use of induced CCG grammar was also evaluated as an extrinsic

[1] Results are announced at http://universaldependencies.org/conll17/results.html

Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies, pages 218–227,
Vancouver, Canada, August 3-4, 2017. © 2017 Association for Computational Linguistics

model in Bisk et al. (2016). They show that although using full CCG derivation trees is superior, CCG lexicon-based grammars also increase performance in a semantics task.

CCG categories have been successfully used in parsing studies as external features and were shown to increase the performance. Note that this use of CCG categories does not allow us to fully access the power of CCG formalism but rather provides a way to use the rich structural information as a means of supertags. Çakıcı (2008) first uses automatically-induced CCG categories from the Turkish treebank as extended (fine) tags (Buchholz and Marsi, 2006) in MacDonald's parser (McDonald et al., 2005). Then they were used for Hindi dependency parsing by Ambati et al. (2013a). Birch et al. (2007) and Nadejde et al. (2017) showed that statistical machine translation benefits from using structurally rich CCG categories (supertags/tags) in the source or target language.

Chen and Manning (2014) proposed a neural network classifier for use in a greedy, transition-based dependency parser. They created a three layer network in which the input layer is fed with word, POS tag and label embeddings, and after the feed forward step, the error is back-propagated to the input layer in order to tune embeddings. They randomly initialized the POS tag and label embeddings, however, as pre-trained word vectors, they used a combination of the embeddings in Collobert et al. (2011) and their trained 50 dimensional word2vec vectors. As this parser only learns and uses dense features with word representations, its parsing speed is at least two times faster than its closest opponent while also improving the accuracy by 2% for English and Chinese. Andor et al. (2016) also created a transition-based dependency parser based on neural networks and word embeddings after the Chen and Manning (2014) work. They proposed to use global instead of local model normalizations to overcome label bias problem with feed forward neural networks. Their parser achieved higher accuracies than former studies in English, Chinese and some other languages as stated in their results.

Dozat and Manning (2016) created a neural network oriented graph based dependency parser. They used bi-affine classifiers to predict arcs and labels. They achieved state-of-the-art or competent accuracies on graph-based parsing for six languages. They improved LAS and UAS score by 1% from previous most accurate graph based parser.

Transition-based dependency parser created by (Kuncoro et al., 2016) is performing as current state-of-the-art with using recurrent neural network grammars. They also outperform most performing graph based parser with increasing attachment scores by almost 2% for English.

3 Method

Gold-standard CCG categories do not exist for languages except a few ones. In order to explore the effects of CCG supertagging on multilingual universal dependencies we assign simplified CCG-based supertags to the multilingual data by using dependency relations from English CCGbank (Hockenmaier and Steedman, 2007). We label training and development data sets for different languages using a tagger trained on English supertagged data and then we use the supertagged training and development data sets for each language in training CCG-based supertaggers and dependency parsers for test data sets in UD treebanks.

Figure 1 shows the overall system that we use for the multilingual universal dependency parsing task (Zeman et al., 2017). We train separate models for dependency parsing for each language.

Section 3.1 explains how we transfer CCG-based supertags to UD training and development data sets, Section 3.2 explains how we train a sequence tagger for assigning supertags to test data and finally Section 3.3 explains how the dependency parser is trained using these supertags.

3.1 Assigning CCG Categories Using Dependency Relations

Dependency relations and predicate-argument structures encoded in CCG categories are parallel most of the time, even though the parent-child directions are different in many cases. Figure 2 shows a sentence from PTB (and CCGbank) with dependency relations above the sentence and predicate-argument relations below the sentence. Many of the edges are symmetric in dependency relations and predicate-argument structure derived from the lexical categories. Figure 3 shows an enlarged view of a part of the sentence in Figure 2 in order to make the labels clearer.

In order to supertag the data, first, the CCGbank

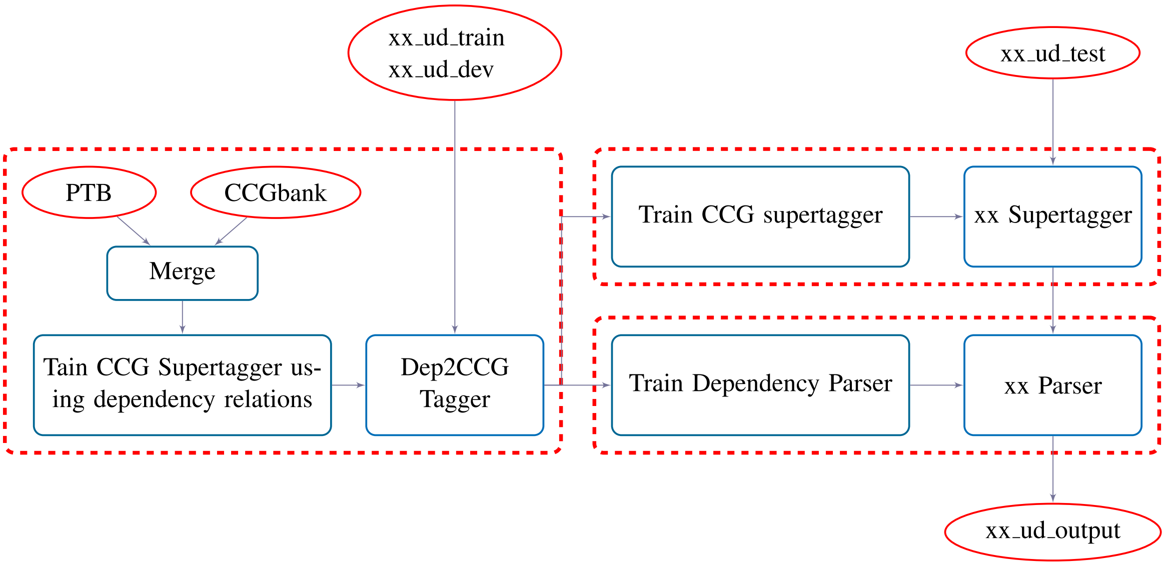

Figure 1: System overview

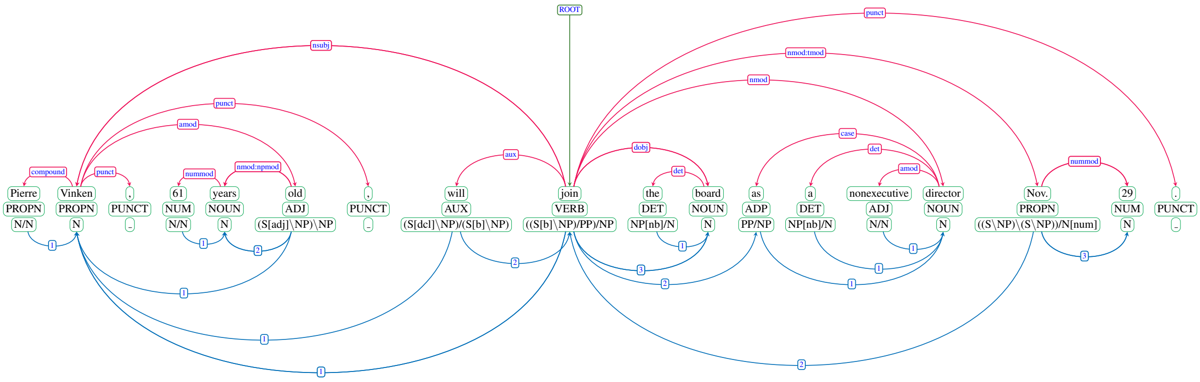

Figure 2: Example: wsj_0001.1

(Hockenmaier and Steedman, 2007) and the Penn Treebank (Marcus et al., 1993) are merged by converting PTB to CONLLU format using Stanford CoreNLP tool (Manning et al., 2014). We then aligned the tokens in both data sets so that CCG categories from CCGbank can be transfered to the MISC field in the CONLLU data.

CCG category	(S[dcl]\NP)/(S[b]\NP)
Without subcategories	(S\NP)/(S\NP)
Without directions	(S-NP)-(S-NP)

Table 1: Category simplification for supertags

A logistic regression classifier is trained to label the tokens in training and development sets using Scikit-Learn (Pedregosa et al., 2011). The classifier only uses universal part-of-speech tags and dependency relations as its features since other features are not common among languages. We use an overly simplified version of CCG categories by removing directional information and features from the CCG supertags. This over-generalization results in decreasing the supertagging accuracy, however, it is necessary so that the English dependency directional information is not imposed on other languages with different word orders. Table 1 shows the simplification of a CCG category

	CCG category	Simplified category
en	(S\NP)\(S\NP)	(S-NP)-(S-NP)
tr	(S\NP[nom])/(S\NP[nom])	(S-NP)-(S-NP)

Table 2: Two adverb categories that take a verb as their arguments in Turkish and English

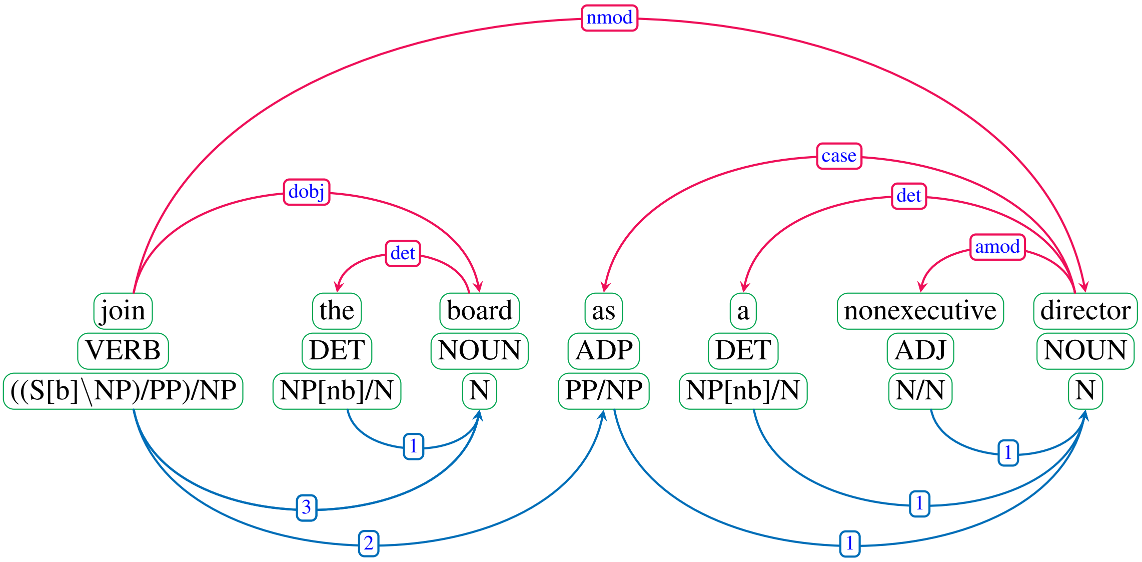

Figure 3: Example: A part of wsj_0001.1

to create a supertag. In Table 2, two adverbs are shown with different CCG categories in two languages. Note that the directions of the slashes in the original forms are different due to different word-order in these languages..

We use dependency relation labels between the word, its head, the head of the head, its dependents and the dependents of its head as features in the tagger. We also use universal part of speech tags for each one of the mentioned words. In English, the only language for which we have the gold standard CCG tags and dependency relations aligned, the re-tagging accuracy is 91.78%. If we additionally use extended part-of-speech tags and CCG argument positions as labels relative to (before or after) the current word, the accuracy increases to 93.55% on English for CCG categories with directional information. However, these are not universal in all languages, so we drop these features. Table 3 shows the features that are extracted for a sample word.

We run this tagger, trained on English CCG data, on every language to generate training and development files with CCG-based supertags for training supetaggers and dependency parsers that will work on test data.

3.2 CCG Sequence Tagging for Training

A supertagger for each language is built using CCG-based supertags transferred with dependency relations. This tagger is a CRF model (Lafferty et al., 2001) built using CRFSuite (Okazaki, 2007). The features are word, part-of-speech tags, key-value pairs in FEATS columns, prefixes and suffixes up to 5 characters, previous and following words and POS tags in a 2-token window. The CRF supertagger on CCGbank with these features (trained only on sections 02-05, tested on section 00) has an accuracy of 93.74%. Lewis and Steedman (2014) give 91.3%, Curran et al. (2007) give 92.6% accuracy with their maximum entropy tagger.

KEY	VALUE	KEY	VALUE
idx	16	pos	NOUN
head_pos	VERB	head_rel	dobj
h_head_pos	VERB	h_head_rel	dobj-ccomp
dep_count	4	h_dep_count	3
dep_1_pos	DET	dep_1_rel	det
dep_2_pos	ADJ	dep_2_rel	amod
dep_3_pos	NOUN	dep_3_rel	nmod
dep_4_pos	NOUN	dep_4_rel	nmod
h_dep_1_pos	NOUN	h_dep_1_rel	nsubj
h_dep_2_pos	AUX	h_dep_2_rel	aux
h_dep_3_pos	NOUN	h_dep_3_rel	dobj

Table 3: Features used in the tagger

Table 4 show features extracted for the word “çalışacak” in sentence “İnşaatta çalışacak vasıfsız işçilerin ...”

KEY	VALUE	KEY	VALUE
Aspect	Perf	Mood	Ind
Polarity	Pos	Tense	Fut
VerbForm	Part	hyp	False
num	False	pnc	False
upp	False	prefix_1	ç
prefix_2	ça	prefix_3	çal
prefix_4	çalı	prefix_5	çalış
suffix_1	k	suffix_2	ak
suffix_3	cak	suffix_4	acak
w	çalışacak	w+1	vasıfsız
w+2	işçilerin	w-1	İnşaatta
w-2	word-1	p	VERB
p+1	ADJ	p+2	NOUN
p-1	NOUN	p-2	POS-1
x	Verb	x+1	Adj
x+2	Noun	x-1	Noun
x-2	XPOS-1		

Table 4: Features used in the tagger

The tagger tries to recover the supertags that are assigned using the dependency relations. On English, the recovery accuracy is 87.36%. Table 5 shows the accuracy of the supertagger on several languages. Note that these are not based on the correct CCG categories but on the assigned supertags via the tagger explained on Section 3.1. This tagger is used on test data preprocessed by UDpipe (Straka et al., 2016) with POSTAGS and FEATS. Then the tagged test data is passed to the dependency parser.

Language	CCG Accuracy
ar	83.47
el	88.90
en	87.36
es	89.70
de	86.45
fr	91.41
he	86.55
tr	77.97
zh	83.83

Table 5: CCG tagger accuracy on several languages

3.3 Dependency Parsing

Two different dependency parsers are experimented with in this study which are powered by two different techniques. First, CCG-based supertags are integrated into the maximum-spanning tree parser ()MSTParser) (McDonald et al., 2005; McDonald, 2006) which is known as the first high-performing graph-based dependency parser. This parser uses discrete local features as input, and thus, the supertags are directly added to this set of features. This implementation shows us the effect of supertagging in a system where the similarities between the supertag groups are not captured semantically. The following parameters are used in all experiments unless otherwise stated:
$order = 2$,
$loss\ type = no\ punctuation$,
$decode\ type = projective$.

We use the Chen and Manning (2014) neural network-based dependency parser to observe how similarities in our supertags affect the model. This parser has been chosen as a baseline as it is the pioneer in using word embeddings for all the features in parsing process. Furthermore, it was given as the parser in the baseline system in Straka et al. (2016) that we compare our results with. In our parser CCG tags are represented with 100 dimensional vectors instead of discrete features. The supertags are obtained from the CCG-based supertagger described in the previous section. In experiments, the following parameters are used: $embedding\ size = 100$ and $hidden\ layer\ size = 200$. For word embeddings, pre-trained 100 dimensional word embeddings by Ginter et al. (2017) are used. For POS tags and supertags, vectors are initialized randomly and are fed to the neural network during training.

4 Results

First, we present our experiments before the release of the test data, then we present the results on the shared task.

4.1 Pre-evaluation

In MSTParser pre-evaluation experiments, we use the Penn Treebank Wall Street Journal segmentation split as sections 2-21 for training and section 23 as the test set. Extra training parameters are as the following:
$training - k = 5$
$loss - type = nopunc$
$decode - type = proj$
$order = 2$
unless stated otherwise in the results. Detailed

descriptions of these parameters can be found in McDonald et al. (2006). These parameters reproduce similar results as McDonald and Pereira (2006) which we use as a baseline to compare our improvements. The version in which the CCG supertags were added also uses the same configuration. Table 6 shows labelled and unlabelled accuracies in detail in these experiments. Accuracy results are obtained from the evaluator built in MSTParser itself. Training and testing times were similar throughout different experiments explained here.

Parser Configuration	UAS	LAS
MSTParser (1st order)	90.7	87.6
MSTParser (2nd order)	91.4	88.3
- xpos (2nd order)	91.1	88.0
- xpos + CCG tags (2nd order)	94.3	90.5
+ CCG tags (2nd order)	94.5	90.8

Table 6: Accuracy on MSTParser with CCG supertags in English (pre-evaluation)

In the pre-evaluation Chen and Manning (2014)'s parser experiment, we also use the Penn Treebank for English as sections 2-21 for training, section 22 as development and section 23 as the test set. For word embedding file, GloVe 50 dimensional data is used (Pennington et al., 2014). Extra configurational parameters are:
$-maxIter$: 20000
$-trainingThreads$: 10
$-embeddingSize$: 50
where $maxIter$ stands for maximum iteration step in neural network training, $trainingThreads$ for number of threads to use during training and $embeddingSize$ for embedding vector size for words, POS tags and supertags.

Reproduction of the results from the original study and our results with our supertags are given in Table 7. These results are obtained by the evaluation method of the original parser.

Also, in the pre-evaluation phase, we test our Chen and Manning (2014) parser-based system on Turkish, German and French data. Shared task-provided data is used for training and development purposes. Word embeddings are used as 100 dimensional vectors from Ginter et al. (2017). Except this difference, all other configuration parameters are the same as in English experiments. Table 8 shows the results. As we see in the results, the tagger predicting the French supertags transferred from English data performs well since the two languages are similar. Also, German tags are inferrable as it is also close to English in grammatical structure. On the other hand, this is not relevant for Turkish, as grammatically, the two languages are quite different. Word order is one of the major differences between these two languages that might have affected the results.

Parser Configuration	UAS	LAS
Chen & Manning	91.3	89.8
Chen & Manning + CCG supertags	95.7	94.6

Table 7: Accuracy on (Chen and Manning, 2014) parser with CCG tags in English (pre-evaluation)

Language	pre-UAS	pre-LAS	post-UAS	post-LAS
Turkish	74.0	61.1	65.8	54.9
German	90.3	80.7	85.4	76.9
French	90.7	75.0	88.1	73.5

Table 8: Accuracy on the Chen and Manning (2014) parser with CCG-based supertags in other languages (pre-evaluation). pre-UAS and pre-LAS stands for accuracies obtained with Chen & Manning, while post-UAS and post-LAS with Chen & Manning + Supertags.

Language	Shared Task Baseline	Our Result
English	82.25	73.40
Spanish	76.76	77.54
Turkish	52.06	48.53
German	71.53	67.97
French	74.57	77.48

Table 9: Our CONLL 2017 Shared task results vs Shared Task baselines (LAS F1 Score)

4.2 Shared Task

In Table 9, we give our shared task results with baselines. Here, we see a drop in accuracy compared to our pre-evaluation phase. The main differences between the systems are pre-trained embedding source and embedding sizes for words, POS tags and labels. In our development experiments, we believe, we are able to capture similarities between POS tags and supertags more efficiently as the embedding size is smaller and the cardinalities of these groups are quite low. This also applies to word embeddings. Other than this, the training iteration count remains the same during experiments. This may be required as we increase the number of features and it needs to be tuned.

For each of the PUD treebanks, we select a model trained on the same language. For some of the languages for which we cannot train a parser, either due to lack of training data or word vectors, and also for the surprise languages, we selected a similar language in the same language family. Table 10 shows the model assignments for the unknown languages.

Language	Selected	Language	Selected
bxr	ru	kmr	fa
sme	fi	hsb	cs
ar_pud	ar	cs_pud	cs
de_pud	de	en_pud	en
es_pud	es	fi_pud	fi
fr_partut	fr	fr_pud	fr
ga	en	gl_treegal	gl
got	de	hi_pud	hi
it_pud	it	ja_pud	ja
kk	tr	la	la_ittb
pt_pud	pt	ru_pud	ru
sl_sst	sl	sv_pud	sv
tr_pud	tr	ug	tr
uk	ru		

Table 10: Selected models for surprise languages and ones without models on training data

Table 11 shows the LAS score of the system on different categories of treebanks as reported in the Shared task paper. All treebanks are shown with the official macro-averaged LAS F1 score. As expected, the system performs better on big treebanks where there are more data instances. PUD treebanks have a big treebank in the same language, therefore the results are close. The difference between our system and the best performing ones is bigger for the small treebanks. The reason for this is that we only train on the language dataset itself and do not use data from other languages for the small treebanks. This causes sparsity issues with the supertagger and the parser. We try to use model trained on a similar language for the surprise languages, however this does not result in a reasonable accuracy since the lexicons are usually very different even if the languages are from the same family and the supertagging relies mostly on the lexical entries and features extracted from them since we do not have the dependency information while decoding. POS tagging error that propagates through the pipeline also affects the performance.

Language Set	LAS F1 Score
All treebanks	61.98
Big treebanks only	68.77
PUD treebanks only	65.30
Small treebanks only	30.84
Surprise languages only	19.39

Table 11: Test results of different categories

Table 12 shows the results of our system on TIRA (Potthast et al., 2014) evaluations of UD test data for each language (Nivre et al., 2017a).

5 Conclusion and Future Work

We experimented with the effects of introducing CCG-based supertags on multilingual universal dependency parsing by taking a radical approach of transferring CCG categories from English to other languages. We used the similarities in dependency formalism and the universal POS tags in order to create CCG lexicons for each language included in the shared task. Since this is a diverse set of languages from different language families and different structural and orthographic properties this transferring is not ideal for many languages.

The existing CCG lexicons for languages such as Turkish were not used for the task since the universal dependency release of Turkish and the dependency treebank the Turkish CCG lexicon were induced from are not aligned on word/tokenization level. Therefore we could not provide accuracy on that data.

We hypothesise that using CCG lexicons from a different language family, especially one with a different word order, may increase the performance of the supertaggers since English only covers a small subset of syntactic properties in a diverse set of languages.

Language	LAS F1 Score	Language	LAS F1 Score
ar_pud	42.68	ar	63.81
bg	82.07	bxr	18.18
ca	82.43	cs_cac	79.52
cs_cltt	68.71	cs_pud	77.06
cs	79.89	cu	60.33
da	68.09	de_pud	65.94
de	67.97	el	76.71
en_lines	70.36	en_partut	72.38
en_pud	75.97	en	73.40
es_ancora	80.89	es_pud	75.26
es	77.54	et	52.08
eu	59.01	fa	75.73
fi_ftb	72.13	fi_pud	69.47
fi	66.46	fr_partut	71.50
fr_pud	71.15	fr_sequoia	75.96
fr	77.48	ga	13.26
gl_treegal	42.91	gl	75.25
got	20.05	grc_proiel	60.08
grc	49.11	he	55.32
hi_pud	49.63	hi	84.54
hr	71.39	hsb	19.59
hu	52.40	id	68.83
it_pud	80.99	it	83.01
ja_pud	74.69	ja	70.62
kk	16.48	kmr	15.86
ko	64.25	la_ittb	70.90
la_proiel	52.23	la	17.21
lv	56.34	nl_lassysmall	71.12
nl	64.73	no_bokmaal	78.51
no_nynorsk	76.19	pl	76.40
pt_br	83.17	pt_pud	68.97
pt	76.48	ro	76.75
ru_pud	65.81	ru_syntagrus	80.88
ru	71.40	sk	70.96
sl_sst	38.00	sl	78.14
sme	23.95	sv_lines	71.93
sv_pud	67.38	sv	74.06
tr_pud	29.24	tr	48.53
ug	15.66	uk	31.68
ur	75.40	vi	36.51
zh	53.84		

Table 12: Test results

After a manual alignment and tagging procedure, the Turkish data can be used in both training and evaluation. We can also group the languages and use similar families of languages to train a common system for them in the future.

One important addition to future work is to induce the CCG supertags in each language including the smaller datasets similar to the approaches used in (Ambati et al., 2013b,a; Çakıcı, 2008) and use these tags in our experiments. We believe adding the specific directional information in CCG categories will help in making more use of the information in the potentially very rich supertags.

Combining two or more CCG lexicons and tagging with the combined model might also be an interesting experiment.

For parsing, we plan to experiment with different word embedding sizes and tune the deep learning parameters. Other than these, we will experiment over neural networks integrated into graph-based dependency parsers. In future, we are planning to use pre-trained POS and CCG tag embeddings in our experiments. If these embeddings can be extracted from corpora on all available tags, this will reduce training time and increase parsing accuracy.

References

Bharat Ram Ambati, Tejaswini Deoskar, and Mark Steedman. 2013a. Using CCG categories to improve hindi dependency parsing. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, ACL 2013, 4-9 August 2013, Sofia, Bulgaria, Volume 2: Short Papers*. pages 604–609.

Bharat Ram Ambati, Tejaswini Deoskar, and Mark Steedman. 2013b. Using ccg categories to improve hindi dependency parsing. In *ACL (2)*. pages 604–609.

Daniel Andor, Chris Alberti, David Weiss, Aliaksei Severyn, Alessandro Presta, Kuzman Ganchev, Slav Petrov, and Michael Collins. 2016. Globally normalized transition-based neural networks. *arXiv preprint arXiv:1603.06042* .

Michael Auli and Adam Lopez. 2011. Efficient ccg parsing: A* versus adaptive supertagging. In Dekang Lin, Yuji Matsumoto, and Rada Mihalcea, editors, *ACL '11: Proceedings of the 49th Annual Meeting on Association for Computational Linguistics*. The Association for Computer Linguistics, pages 1577–1585.

Srinivas Bangalore and Aravind K Joshi. 1999. Supertagging: An approach to almost parsing. *Computational linguistics* 25(2):237–265.

Alexandra Birch, Miles Osborne, and Philipp Koehn. 2007. Ccg supertags in factored statistical machine translation. In *Proceedings of the Second Workshop on Statistical Machine Translation*. Association for Computational Linguistics, pages 9–16.

Yonatan Bisk, Siva Reddy, John Blitzer, Julia Hockenmaier, and Mark Steedman. 2016. Evaluating induced ccg parsers on grounded semantic parsing. In *EMNLP*.

Sabine Buchholz and Erwin Marsi. 2006. CoNLL-X shared task on multilingual dependency parsing. In *Proceedings of the 10th Conf. on Computational Natural Language Learning (CoNLL-X)*. SIGNLL.

Ruket Çakıcı. 2008. *Wide-Coverage Parsing for Turkish*. Ph.D. thesis, University of Edinburgh.

Danqi Chen and Christopher D Manning. 2014. A fast and accurate dependency parser using neural networks. In *EMNLP*. pages 740–750.

Stephen Clark and James R. Curran. 2004. The importance of supertagging for wide-coverage ccg parsing. In *Proceedings of the 20th International Conference on Computational Linguistics*. Association for Computational Linguistics, Stroudsburg, PA, USA, COLING '04. https://doi.org/10.3115/1220355.1220396.

Stephen Clark and James R. Curran. 2007. Wide-coverage efficient statistical parsing with CCG and log-linear models. *Computational Linguistics* 33(4).

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *Journal of Machine Learning Research* 12(Aug):2493–2537.

Timothy Dozat and Christopher D Manning. 2016. Deep biaffine attention for neural dependency parsing. *arXiv preprint arXiv:1611.01734* .

Agnieszka Falenska, Anders Bjorkelund, Ozlem Cetinoglu, and Wolfgang Seeker. 2015. Stacking or supertagging for dependency parsing what's the difference? In *Proceedings of the 14th International Conference on Parsing Technologies*. Association for Computational Linguistics, Bilbao, Spain, pages 118–129. http://www.aclweb.org/anthology/W15-2215.

Kilian Foth, Tomas By, and Wolfgang Menzel. 2006. Guiding a constraint dependency parser with supertags. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, pages 289–296.

Filip Ginter, Jan Hajič, Juhani Luotolahti, Milan Straka, and Daniel Zeman. 2017. CoNLL 2017 shared task - automatically annotated raw texts and word embeddings. LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics, Charles University. http://hdl.handle.net/11234/1-1989.

Julia Hockenmaier and Mark Steedman. 2002. Generative models for statistical parsing with combinatory categorial grammar. In *ACL '02: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*. pages 335–342.

Julia Hockenmaier and Mark Steedman. 2007. Ccgbank: a corpus of ccg derivations and dependency structures extracted from the penn treebank. *Computational Linguistics* 33(3):355–396.

Adhiguna Kuncoro, Miguel Ballesteros, Lingpeng Kong, Chris Dyer, Graham Neubig, and Noah A Smith. 2016. What do recurrent neural network grammars learn about syntax? *arXiv preprint arXiv:1611.05774* .

John Lafferty, Andrew McCallum, Fernando Pereira, et al. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the eighteenth international conference on machine learning, ICML*. volume 1, pages 282–289.

Mike Lewis and Mark Steedman. 2014. A* CCG parsing with a supertag-factored model. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*. pages 990–1000.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*. pages 55–60. http://www.aclweb.org/anthology/P/P14/P14-5010.

Mitchell P Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. 1993. Building a large annotated corpus of english: The penn treebank. *Computational linguistics* 19(2):313–330.

Ryan McDonald. 2006. *Discriminative Learning and Spanning Tree Algorithms for Dependency Parsing*. Ph.D. thesis, University of Pennsylvania.

Ryan McDonald, Koby Crammer, and Fernando Pereira. 2005. Online large-margin training of dependency parsers. In *Proceedings of ACL 2005*. Ann Arbor, MI, USA.

Ryan McDonald et al. 2006. Mstparser readme. http://www.seas.upenn.edu/~strctlrn/MSTParser/README.

Ryan T McDonald and Fernando CN Pereira. 2006. Online learning of approximate dependency parsing algorithms. In *EACL*. pages 81–88.

Maria Nadejde, Siva Reddy, Rico Sennrich, Tomasz Dwojak, Marcin Junczys-Dowmunt, Philipp Koehn, and Alexandra Birch. 2017. Syntax-aware neural machine translation using ccg. *arXiv preprint arXiv:1702.01147* .

Joakim Nivre, Željko Agić, Lars Ahrenberg, et al. 2017a. Universal dependencies 2.0 CoNLL 2017 shared task development and test data. LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics, Charles University. http://hdl.handle.net/11234/1-2184.

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajič, Christopher Manning, Ryan McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, Reut Tsarfaty, and Daniel Zeman. 2016. Universal Dependencies v1: A multilingual treebank collection. In *Proceedings of the 10th International Conference on Language Resources and Evaluation (LREC 2016)*. European Language Resources Association, Portoro, Slovenia, pages 1659–1666.

Joakim Nivre et al. 2017b. Universal Dependencies 2.0. LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics, Charles University, Prague, `http://hdl.handle.net/11234/1-1983`. http://hdl.handle.net/11234/1-1983.

Naoaki Okazaki. 2007. Crfsuite: a fast implementation of conditional random fields (crfs). http://www.chokkan.org/software/crfsuite/.

Hiroki Ouchi, Kevin Duh, and Yuji Matsumoto. 2014. Improving dependency parsers with supertags. In *EACL*. pages 154–158.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research* 12:2825–2830.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*. pages 1532–1543. http://www.aclweb.org/anthology/D14-1162.

Martin Potthast, Tim Gollub, Francisco Rangel, Paolo Rosso, Efstathios Stamatatos, and Benno Stein. 2014. Improving the reproducibility of PAN's shared tasks: Plagiarism detection, author identification, and author profiling. In Evangelos Kanoulas, Mihai Lupu, Paul Clough, Mark Sanderson, Mark Hall, Allan Hanbury, and Elaine Toms, editors, *Information Access Evaluation meets Multilinguality, Multimodality, and Visualization. 5th International Conference of the CLEF Initiative (CLEF 14)*. Springer, Berlin Heidelberg New York, pages 268–299. https://doi.org/10.1007/978-3-319-11382-1_22.

Mark Steedman. 2000. *The Syntactic Process*. MIT Press, Cambridge, MA.

Milan Straka, Jan Hajič, and Jana Straková. 2016. UDPipe: trainable pipeline for processing CoNLL-U files performing tokenization, morphological analysis, POS tagging and parsing. In *Proceedings of the 10th International Conference on Language Resources and Evaluation (LREC 2016)*. European Language Resources Association, Portoro, Slovenia.

Daniel Zeman, Martin Popel, Milan Straka, Jan Hajič, Joakim Nivre, Filip Ginter, Juhani Luotolahti, Sampo Pyysalo, Slav Petrov, Martin Potthast, Francis Tyers, Elena Badmaeva, Memduh Gökırmak, Anna Nedoluzhko, Silvie Cinková, Jan Hajič jr., Jaroslava Hlaváčová, Václava Kettnerová, Zdeňka Urešová, Jenna Kanerva, Stina Ojala, Anna Missilä, Christopher Manning, Sebastian Schuster, Siva Reddy, Dima Taji, Nizar Habash, Herman Leung, Marie-Catherine de Marneffe, Manuela Sanguinetti, Maria Simi, Hiroshi Kanayama, Valeria de Paiva, Kira Droganova, Hěctor Martínez Alonso, Hans Uszkoreit, Vivien Macketanz, Aljoscha Burchardt, Kim Harris, Katrin Marheinecke, Georg Rehm, Tolga Kayadelen, Mohammed Attia, Ali Elkahky, Zhuoran Yu, Emily Pitler, Saran Lertpradit, Michael Mandl, Jesse Kirchner, Hector Fernandez Alcalde, Jana Strnadova, Esha Banerjee, Ruli Manurung, Antonio Stella, Atsuko Shimada, Sookyoung Kwak, Gustavo Mendonça, Tatiana Lando, Rattima Nitisaroj, and Josie Li. 2017. CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*. Association for Computational Linguistics.