Semantic Tagging with Deep Residual Networks

Johannes Bjerva

Center for Language and Cognition Groningen, University of Groningen
Department of Linguistics, Stockholm University
j.bjerva@rug.nl, johannes.bjerva@ling.su.se

Abstract

We propose a novel semantic tagging task, *semtagging*, tailored for the purpose of multilingual semantic parsing, and present the first tagger using deep residual networks (ResNets). Our tagger uses both word and character representations.We evaluate the tagset both intrinsically on the new task of semantic tagging, as well as on Part-of-Speech (POS) tagging. Our system, consisting of a ResNet and an auxiliary loss function predicting our semantic tags, significantly outperforms prior results on English Universal Dependencies POS tagging (95.71% accuracy on UD v1.2 and 95.67% accuracy on UD v1.3).

1. Introduction

A key issue in computational semantics is the transferability of semantic information across languages. Many semantic parsing systems depend on sources of information such as POS tags (Pradhan et al., 2004; Copestake et al., 2005; Bos, 2008; Butler, 2010; Berant and Liang, 2014). However, these tags are often customised for the language at hand (Marcus et al., 1993) or massively abstracted, such as the Universal Dependencies tagset (Nivre et al., 2016). Furthermore, POS tags are syntactically oriented, and therefore often contain both irrelevant and insufficient information for semantic analysis and deeper semantic processing. This means that, although POS tags are highly useful for many downstream tasks, they are unsuitable for semantic parsing.

We present a novel set of semantic labels tailored for the purpose of multilingual semantic parsing. This tagset (i) abstracts over POS and named entity types; (ii) fills gaps in semantic modelling by adding new categories (for instance for phenomena like negation, modality, and quantification); and (iii) generalises over specific languages. We introduce and motivate this new task in this paper, and refer to it as *semantic tagging*. Our experiments aim to answer the following two research questions: i) Can we outperform off-the-shelf sequence taggers by using recent neural network architectures?; ii) Can we find evidence that semtags are effective for NLP tasks other than semantic parsing?

To address the first question, we will look at convolutional neural networks (CNNs) and recurrent neural networks (RNNs). A recent development is the emergence of deep residual networks (ResNets), a building block for CNNs. ResNets consist of several stacked residual units, which can be thought of as a collection of convolutional layers coupled with a 'shortcut' which aids the propagation of the signal in a neural network. This allows for the construction of much deeper networks, since keeping a 'clean' information path in the network facilitates optimisation (He et al., 2016). ResNets have recently shown state-of-the-art performance for image classification tasks (He et al., 2015; He et al., 2016), and have also seen some recent use in NLP (Östling, 2016; Conneau et al., 2016; Bjerva et al., 2016; Bjerva, 2016).

To answer our second question, we carry out an extrinsic evaluation. We investigate the effect of using semantic tags as an auxiliary task for POS tagging. Since POS tags are useful for many NLP tasks, it follows that semantic tags must be useful if they can improve POS tagging.

2. Semantic Tagging

2.1 Background

We refer to *semantic tagging*, or *semtagging*, as the task of assigning semantic class categories to the smallest meaningful units in a sentence (i.e. words in the context of this paper). Semantic tagging reaches all parts of speech. Examples where semantic classes disambiguate are reflexive versus emphasising pronouns (both POS tagged as `PRP`, personal pronoun); the comma, that could be a conjunction, disjunction, or apposition; intersective vs. subsective and privative adjectives (all POS tagged as `JJ`, adjective); proximal vs. medial and distal demonstratives (see Example 1); subordinate vs. coordinate discourse relations; agent nouns vs. entity nouns. The set of semantic tags that we use in this paper is established in a data-driven manner, considering four languages in a parallel corpus (English, German, Dutch and Italian). This first inventory of classes comprises 13 coarse-grained tags and 66 fine-grained tags. The tagset also includes named entity classes (see Example 2).

(1) *These cats live in that house .*
`PRX CON ENS REL DST CON NIL`

(2) *Ukraine 's glory has not yet perished*
`GPE HAS CON ENT NOT IST EXT`
, neither her freedom .
`NIL NOT HAS CON NIL`

In Example 1, both *these* and *that* would be tagged as `DT`. However, with our semantic tagset, they are disambiguated as `PRX` (proximal) and `DST` (distal). In Example 2, *Ukraine* is tagged as `GPE` rather than `NNP`.

For further description of the semantic tags, and an overview of all tags used, we refer the reader to Bjerva et al. (2016).

2.2 Annotated Data

We use two semtag datasets. The Groningen Meaning Bank (GMB) corpus of English texts (1.4 million words) contain-

ing silver standard semantic tags obtained by running a simple rule-based semantic tagger (Bos et al., Forthcoming).

Our second dataset is smaller but equipped with gold standard semantic tags and used for testing (PMB, the Parallel Meaning Bank). It comprises a selection of 400 sentences of the English part of a parallel corpus. It has no overlap with the GMB corpus. The semantic tags were obtained as for the GMB, and then corrected by a human annotator.

For the extrinsic evaluation, we use the POS data in the English portion of the Universal Dependencies dataset, version 1.2 and 1.3 (Nivre et al., 2016).

3. Method

Our tagger is a hierarchical deep neural network consisting of a bidirectional Gated Recurrent Unit (GRU) network a the upper level, and a Convolutional Neural Network (CNN) and/or Deep Residual Network (ResNet) at the lower level.

3.1 Gated Recurrent Unit networks

GRUs (Cho et al., 2014) are a recently introduced variant of RNNs, and are designed to prevent vanishing gradients, thus being able to cope with longer input sequences than vanilla RNNs. A bi-directional GRU is a GRU which makes both forward and backward passes over sequences, and can therefore use both preceding and succeeding contexts to predict a tag (Graves and Schmidhuber, 2005).

3.2 Deep Residual Networks

Deep Residual Networks (ResNets) are built up by stacking residual units. ResNets can be intuitively understood by thinking of residual functions as paths through which information can propagate easily. This means that, in every layer, a ResNet learns more complex feature combinations, which it combines with the shallower representation from the previous layer. This architecture allows for the construction of much deeper networks.

3.3 System description

We use pre-trained word embeddings, which are passed directly into a two-layer bi-GRU. We further use CNNs for character-level modelling. The resulting character-based word representations are concatenated with the word embeddings (depending on condition), and passed into the bi-GRU.

Recent work has shown that the addition of an auxiliary loss function can be beneficial to several tasks (Plank et al., 2016; Søgaard and Goldberg, 2016). We experiment with predicting coarse semtags as an auxiliary task for the semtagging experiments. Similarly, we also experiment with POS tagging, where we use the fine semtags as an auxiliary task.

3.4 Hyperparameters

All hyperparameters are tuned with respect to loss on the semtag validation set. We use rectified linear units (ReLUs) for all activation functions (Nair and Hinton, 2010), and apply dropout with $p = 0.1$ to both input weights and recurrent weights in the bi-GRU (Srivastava et al., 2014). In the CNNs, we apply batch normalisation (Ioffe and Szegedy, 2015). In our basic CNN, we apply a 4×8 convolution, followed by 2×2 maximum pooling, followed by 4×4 convolution and another 2×2 maximum pooling. Our ResNet has the same setup, with the addition of a residual connection. All experiments were run with early stopping monitoring validation set loss, using a maximum of 50 epochs. Optimisation is done using the ADAM algorithm (Kingma and Ba, 2014), with the categorical cross-entropy loss function. In our experiments, we weight the auxiliary loss with $\lambda = 0.1$, as set on the semtag auxiliary task.

4. Evaluation

We evaluate our tagger on two tasks: semantic tagging (ST) and POS tagging. Note that the tagger is developed solely on the semantic tagging task, using the GMB silver training and validation data (i.e. no fine-tuning of hyperparameters for POS tagging). We calculate significance using bootstrap resampling (Efron and Tibshirani, 1994). We manipulate the following independent variables in our experiments: i) character and word representations ($\vec{w}, \vec{c}$); ii) convolutional representations (Basic CNN and ResNets); iii) auxiliary loss (using coarse semtags on ST and fine semtags on UD).

We compare our results to four baselines: i) the most frequent baseline per word (MFC); ii) the trigram statistic based TNT tagger (Brants, 2000); iii) the BI-LSTM baseline, running the off-the-shelf state-of-the-art POS tagger for the UD dataset (Plank et al., 2016) (default parameters with pre-trained Polyglot embeddings (Al-Rfou et al., 2013)); iv) we use a baseline consisting of running our own system with only a BI-GRU using word representations ($\vec{w}$) from pre-trained Polyglot embeddings.

4.1 Experiments on semantic tagging

We evaluate our system on our silver semtag dataset and our gold semtag dataset. For the +AUX condition we use coarse semtags as an auxiliary loss. Results from these experiments are shown in Table 1.

4.2 Experiments on Part-of-Speech tagging

We evaluate our system on v1.2 and v1.3 of the English part of the Universal Dependencies (UD) data. We report results for POS tagging alone, comparing to commonly used baselines and prior work using LSTMs, as well as using the fine-grained semantic tags as auxiliary information. For the +AUX condition, we train a single joint model using a multi-task objective, with POS and ST as our two tasks. This model is trained on the concatenation of the ST silver data with the UD data, updating the loss of the respective task of an instance in each iteration. Results from these experiments are shown in Table 2.

5. Discussion

5.1 Performance on semantic tagging

The overall best system is the ResNet combining both word and character representations $\vec{c} \wedge \vec{w}$. It outperforms all baselines, including the recently proposed RNN-based bi-LSTM. On the ST silver data, a significant difference ($p < 0.01$) is found when comparing our best system to the strongest baseline (BI-LSTM). On the ST gold data, we

	BASELINES				BASIC CNN			RESNET		
	MFC	TNT	BI-LSTM	BI-GRU	$\vec{c}$	$\vec{c} \wedge \vec{w}$	+AUX	$\vec{c}$	$\vec{c} \wedge \vec{w}$	+AUX
Semtag Silver	84.64	92.09	94.98	94.26	91.39	94.63	94.53	94.39	**95.14**	94.23
Semtag Gold	77.39	80.73	82.96	80.26	69.21	76.83	80.73	76.89	**83.64**	74.84

Table 1: Experiment results on semtag (ST) test sets (% accuracy). MFC indicates the per-word most frequent class baseline, TNT indicates the TNT tagger, and BI-LSTM indicates the system by Plank et al. (2016). BI-GRU indicates the $\vec{w}$ only baseline. $\vec{w}$ indicates usage of word representations, $\vec{c}$ indicates usage of character representations. The +AUX column indicates the usage of an auxiliary loss.

	BASELINES				BASIC CNN			RESNET		
	MFC	TNT	BI-LSTM	BI-GRU	$\vec{c}$	$\vec{c} \wedge \vec{w}$	+AUX	$\vec{c}$	$\vec{c} \wedge \vec{w}$	+AUX
UD v1.2	85.06	92.66	95.17	94.39	77.63	94.68	95.19	92.65	94.92	**95.71**
UD v1.3	85.07	92.69	95.04	94.32	77.51	94.89	95.34	92.63	94.88	**95.67**

Table 2: Experiment results on Universal Dependencies (UD) test sets (% accuracy).

observe significant differences ($p < 0.0025$). Adding the coarse-grained semtags as auxiliary task only helps for the weaker CNN model.

It is especially noteworthy that the ResNet character-only system outperforms the BI-GRU and TNT baselines, and is considerably better than the basic CNN. Since performance increases further when adding in $\vec{w}$, it is clear that the character and word representations are complimentary in nature. The high results for characters only are particularly promising for multilingual language processing, a direction we want to explore next.

5.2 Performance on POS tagging

Our system was tuned solely on semtag data. This is reflected in, e.g., the fact that even though our $\vec{c} \wedge \vec{w}$ ResNet system outperforms the Plank et al. (2016) system on semtags, we are substantially outperformed on UD 1.2 and 1.3 in this setup. However, adding an auxiliary task based on our semtags markedly increases performance on POS tagging. In this setting, our tagger outperforms the BI-LSTM system, and results in new state-of-the-art results on both UD 1.2 (95.71% accuracy) and 1.3 (95.67% accuracy). The difference between the BI-LSTM system and our best system is significant at $p < 0.0025$.

6. Conclusions

We introduce a semantic tagset tailored for multilingual semantic parsing. We evaluate tagging performance using standard CNNs and the recently emerged ResNets. ResNets are more robust and result in our best model. Combining word and ResNet-based character representations helps to outperform state-of-the art taggers on semantic tagging. Coupling this with an auxiliary loss from our semantic tagset yields state-of-the art performance on the UD 1.2 and 1.3 POS datasets.

References

Rami Al-Rfou, Bryan Perozzi, and Steven Skiena. 2013. Polyglot: Distributed word representations for multilingual nlp. *CoNLL-2013*.

Jonathan Berant and Percy Liang. 2014. Semantic parsing via paraphrasing. In *ACL*, pages 1415–1425.

Johannes Bjerva, Barbara Plank, and Johan Bos. 2016. Semantic tagging with deep residual networks. *arXiv preprint arXiv:1609.07053*.

Johannes Bjerva. 2016. Byte-based language identification with deep convolutional networks. *arXiv preprint arXiv:1609.09004*.

Johan Bos, Valerio Basile, Kilian Evang, Noortje Venhuizen, and Johannes Bjerva. Forthcoming. The Groningen Meaning Bank. In Nancy Ide and James Pustejovsky, editors, *The Handbook of Linguistic Annotation*. Springer, Berlin.

Johan Bos. 2008. Wide-Coverage Semantic Analysis with Boxer. In J. Bos and R. Delmonte, editors, *Semantics in Text Processing. STEP 2008 Conference Proceedings*, volume 1 of *Research in Computational Semantics*, pages 277–286. College Publications.

Thorsten Brants. 2000. Tnt: a statistical part-of-speech tagger. In *Proceedings of the sixth conference on Applied natural language processing*, pages 224–231. Association for Computational Linguistics.

Alastair Butler. 2010. *The Semantics of Grammatical Dependencies*, volume 23. Emerald Group Publishing Limited.

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder-decoder for statistical machine translation. In *EMNLP*.

Alexis Conneau, Holger Schwenk, Loïc Barrault, and Yann Lecun. 2016. Very deep convolutional networks for natural language processing. *arXiv preprint arXiv:1606.01781*.

Ann Copestake, Dan Flickinger, Ivan Sag, and Carl Pollard. 2005. Minimal recursion semantics: An introduction. *Journal of Research on Language and Computation*, 3(2–3):281–332.

Bradley Efron and Robert J Tibshirani. 1994. *An introduction to the bootstrap*. CRC press.

Alex Graves and Jürgen Schmidhuber. 2005. Framewise phoneme classification with bidirectional lstm and other neural network architectures. *Neural Networks*, 18(5):602–610.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Identity mappings in deep residual networks. *arXiv preprint arXiv:1603.05027*.

Sergey Ioffe and Christian Szegedy. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*.

Diederik Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

M.P. Marcus, B. Santorini, and M.A. Marcinkiewicz. 1993. Building a Large Annotated Corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330.

Vinod Nair and Geoffrey E Hinton. 2010. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 807–814.

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajic, Christopher D Manning, Ryan McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, et al. 2016. Universal dependencies v1: A multilingual treebank collection. In *Proceedings of the 10th International Conference on Language Resources and Evaluation (LREC 2016)*.

Robert Östling. 2016. Morphological reinflection with convolutional neural networks. In *Proceedings of the 2016 Meeting of SIGMORPHON*, Berlin, Germany. Association for Computational Linguistics.

Barbara Plank, Anders Søgaard, and Yoav Goldberg. 2016. Multilingual part-of-speech tagging with bidirectional long short-term memory models and auxiliary loss. In *Proceedings of ACL 2016, arXiv preprint arXiv:1604.05529*.

Sameer S Pradhan, Wayne Ward, Kadri Hacioglu, James H Martin, and Daniel Jurafsky. 2004. Shallow semantic parsing using support vector machines. In *HLT-NAACL*, pages 233–240.

Anders Søgaard and Yoav Goldberg. 2016. Deep multi-task learning with low level tasks supervised at lower layers. In *The 54th Annual Meeting of the Association for Computational Linguistics*, page 231.

Nitish Srivastava, Geoffrey E Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1):1929–1958.