Using hyperlinks to improve multilingual partial parsers

Anders Søgaard
Dpt. of Computer Science, University of Copenhagen
http://cst.dk/anders/
soegaard@di.ku.dk

Abstract

Syntactic annotation is costly and not available for the vast majority of the world's languages. We show that sometimes we can do away with less labeled data by exploiting more readily available forms of mark-up. Specifically, we revisit an idea from Valentin Spitkovsky's work (2010), namely that hyperlinks typically bracket syntactic constituents or chunks. We strengthen his results by showing that not only can hyperlinks help in low resource scenarios, exemplified here by Quechua, but learning from hyperlinks can also improve state-of-the-art NLP models for English newswire. We also present out-of-domain evaluation on English Ontonotes 4.0.

1 Introduction

Syntactic analysis can be used to improve knowledge extraction, speech synthesis, machine translation, and error correction, for example, but the quality of syntactic parsers relies heavily on the quality and amount of available annotated data. This holds in particular for full syntactic parsing, but even for more robust *partial* parsers, good models require large and representative, annotated corpora.

Such annotated corpora are costly to produce and generally not available for the vast majority of the world's languages. Even for English, resources are limited, and state-of-the-art parsers for English newswire are trained on 30 years old newswire from a single newspaper. When evaluated on more recent newswire or other newspapers, we observe significant performance drops.

This is a combination of overfitting and data scarcity. While more annotated resources can improve this situation, annotation does not seem to scale with our needs for automated syntactic analysis, or with the rapid development of modern languages like English. Hence, we have to consider other types of data to adapt our models to other varieties of newswire, or of language, more generally.

Using (more representative) raw text in combination with (less representative) annotated data to do semi-supervised learning is challenging, but occasionally successful. In this paper, we consider an equally readily available, potential source of weak supervision, namely hypertext. Consider the following hypertext:

```
The violence, which has already been called some
evocative names -- <href>intifada<\ href>,
<href>jihad<\href>, <href>jihad<guerilla
war<\href>, <href>insurrection<\href>,
<href>rebellion<\href>, and <href>civil
war<\href> -- prompts several reflections.
```

This sentence is a random sentence taken from the Internet. The mark-up is hyperlinks, referring the reader to related websites. The hyperlinks mark passsages of the text highlighting the topics of the linked websites.

The marked passages are *intifada, jihad, guerilla war, insurrection, rebellion* and *civil war*. Note that these are not just words, but also phrases. In this example, they are all noun phrases.

Spitkovsky et al. (2010) also looked at hyperlinks and observed that the vast majority of marked passages were syntactic constituents such as noun and verb phrases. He then went on to show that this data is potentially useful for unsupervised induction of dependency parsers.

We build directly on this work, but go on to show that hyperlinks are not just useful for unsupervised induction of NLP models. It is also possible to improve state-of-the-art supervised NLP models, by jointly learning to predict hyperlinks from raw HTML files. Specifically, we show that

Proceedings of the 15th International Conference on Parsing Technologies, pages 67–71,
Pisa, Italy; September 20–22, 2017. ©2017 Association for Computational Linguistics

hard parameter sharing of hidden layers with a deep bi-LSTM model for predicting hyperlinks is an efficient regularizer for several state-of-the-art NLP models.

Contributions Our contributions are as follows: (a) We revisit the idea of using raw HTML data for weak supervision of NLP models. (b) We show that multi-task learning with hyperlink prediction as an auxiliary task improves performance across three tasks: syntactic chunking, semantic supersense tagging, and CCG supertagging. We also see improvements on out-of-domain English data, as well as in experiments with syntactic chunking with hyperlinks for Quechua.

Related work Hard parameter sharing of hidden layers has become a popular approach to multi-task learning. It was originally introduced in Caruana (1993), but first applied in NLP in Collobert et al. (2011), and it was shown, empirically, to be an effective regularizer across two different NLP tasks in Søgaard and Goldberg (2016). Using more readily available data resources that are not annotated by linguists, but still carry linguistic signals, was previously explored by Klerke et al. (2016) and Plank (2016).

Baxter (2000) shows, in the context of linear models, that if two problems, P and R, share optimal hypothesis classes, then the induction of a model from a sample of P can efficiently regularize the induction of a model from a sample of R. This is too strong an assumption for our purposes, obviously, since even our label sets are different, but we also have more wiggle-room than heavily mean-constrained linear models, for example. In fact, hidden layer sharing relaxes the above assumption quite a bit. We do not need the optimal hypothesis classes to overlap. Hidden layer sharing can work even with the optimal hypothesis classes of P and R distinct, if there is a joint representation such that P and R both become linearly separable. Whether this is the case, is an empirical question.

2 Experiments

Model Our model merges two deep recurrent neural networks through hard parameter sharing. We use three-layered, bi-directional long short-term memory networks (LSTMs) (Hochreiter and Schmidhuber, 1997), in a way similar to Søgaard and Goldberg (2016). We optimize hyper-parameters on development data for chunking in a single-task architecture, training for 10 epochs, and using a hidden layer size that is equal to the embedding layer size. For English, we use the SENNA word embedding for English and 50-dimensional hidden layers.[1] For Quechua, we use Polyglot embeddings and 64-dimensional hidden layers.

In a multi-task learning (MTL) setting, we have several prediction tasks over the same input space. In our case, the input is the words in a sentence, and the different tasks are syntactic chunking, semantic supersense tagging and CCG supertagging. And hyperlink prediction. Each task has its own output vocabulary (a task specific tagset), but all of them map any length n input sequence into a length n output sequence.

The most common approach to multi-task learning in NLP these days is to share parameters across most of the hidden layers of two or more single task networks. In the k-layers deep bi-LSTM tagger described above this is naturally achieved by sharing the bi-LSTM part of the network across tasks, but training a specialized classification tagger $f_t(v_i^k)$ for each task t.

Note that this particular kind of multi-task learning can also be cast as a kind of mean-constrained matrix regularization. While in some sense, hard parameter sharing is more heavily regularized than more traditional approaches to multi-task learning, such as mean-constrained L2 regularization, we obtain more wiggle room by only sharing the embedding and LSTM parameters.

Our model is implemented in pyCNN and made available at:

`bitbucket.org/soegaard/hyperlink-iwpt17`

English data In our English in-domain experiments, we use three datasets for our target tasks, namely the Penn Treebank for syntactic chunking (Marcus et al., 1993), the SemCor corpus for semantic supersense tagging (Miller et al., 1994; Ciaramita and Altun, 2006), and the CCGBank[2] for CCG super-tagging. See Figure 1 for an example of all three layers of annotation. The training section of the chunking dataset consists of 8936 sentences. SemCor contains 15465 sentences, and the CCGBank contains 39604 sentences. For our auxiliary task, for replicability (and as a tribute

[1] `http://ronan.collobert.com/senna/`

[2] LDC2005T13

Words:	Are	prairie	dogs	conscious
Chunking:	B-VP	B-NP	I-NP	B-ADJP
Semcor:	O	O	Animal.N	Cognition.ADJ
CCG:	(S[adj]\NP))/NP	NP	NP\NP	S[adj]\NP)
Href	0	B-HREF	I-HREF	O

Figure 1: Examples of linguistic annotation

to Valentin's seminal work), we use the hypertext dump used in Spitkovsky et al. (2010), made publicly available,[3], which contains 2000 sentences. To evaluate the robustness of our syntactic chunker, we also evaluate it across multiple domains using data from Ontonotes 4.0.[4]

Quechua data We use constituent annotations of Quechua sentences, from Rios (2015), and convert them into partial annotations. The sentences are from an autobiography. The training data consists of 1500 sentences, and the test data is 837 sentences. The annotations only provide NP and VP bracketing, leaving us with five labels. For our auxiliary task, we use 350 sentences from Quechua Wikipedia that contain hyperlinks. The data is made publicly available.[5]

Balance between tasks Our auxiliary datasets are relatively small, in the light of hyperlinks being readily available. In hard parameter sharing, it is important not to swamp the main task, and as our learning curve experiments indicate, it would not be beneficial to sample more auxiliary task data. Soft parameter sharing approaches may better leverage large volumes of hyperlink data. See discussion of learning curves in §3.

3 Results

In all our experiments, we report averages over three runs.

English in-sample tests In our first experiment, we train an LSTM on English newswire and apply it to English newswire, using the standard datasets from the English Penn Treebank, Semcor, and the CCGBank. Our baseline is a single-task LSTM architecture, with the hyper-parameters suggested by Søgaard and Goldberg (2016). We verify that this leads to state-of-the-art performance. In fact, our single-task baseline is slightly better than the one used in Søgaard and Goldberg (2016). We then train the same network architecture with the hyperlink data from Spitkovsky et al. (2010) as our auxiliary data.

Using hyperlinks as auxiliary data leads to moderate improvements for syntactic chunking, and very big improvements for supersense tagging and CCG supertagging. Where for syntactic chunking, the error reduction is less than 3%, it is 17% for supersense tagging, and 13% for CCG supertagging.

English out-of-sample tests We use syntactic chunking data from OntoNotes 4.0. The data includes manually annotated data from several sources across several domains: newswire, broadcast, broadcasted news, and weblogs. We use a single file for training (WSJ), and a single file for development (CCTV), and all other files for testing. We have 23 files for testing, spanning weblogs from C2E to (English) news from Xinhua.

Performance is generally much lower, because of the divergence between training and test data. Whereas before, performance (F_1) on test data was about 95%, cross-domain performance is generally about 85%. See results in Table 2. The average gain from multi-task learning remains small, even when we consider the test domains with highest divergence (weblogs).

It is important to note that unlike other experiments using multi-task learning for domain adaptation, e.g., Søgaard and Goldberg (2016), our auxiliary task data is sampled from the domain of the training data (newswire), not of the test data. This may effect results quite a bit, and our results do therefore not contradict the results in Søgaard and Goldberg (2016) and related work.

Also, note that there are other possible explanations for the differences in performance gains across target tasks. One possible predictor for multi-task learning gains may for example be properties of single-task learning curves, variance across model parameters, etc. See Bingel and Søgaard (2017) for work exploring such predictors of when multi-task learning works

[3] nlp.stanford.edu/valentin/pubs/markup-data.tar.bz2

[4] LDC2011T03

[5] bitbucket.org/soegaard/hyperlink-iwpt17

	English			Quechua
	Chunking	SemCor	CCG	Chunking
LSTM	0.9543	0.6757	0.9169	0.7169
LSTM w. HREF	**0.9555**	**0.7312**	**0.9275**	**0.7283**
Err.red.	0.0263	0.1711	0.1276	0.0403
S&G16 (bl)	0.9528	-	0.9104	-
S&G16 (best)	0.9556	-	0.9326	-

Table 1: Improvements in F_1 using hyperlinks as auxiliary data across three NLP tasks. Søgaard and Goldberg (2016) for comparison (S&G16). S&G16 (best) is similar to our hyperlinks model, but uses POS-tag annotated data for co-supervising the initial LSTM layer instead of hyperlinks data for co-supervising *all* the hidden layers. Previous work on SemCor assumes gold-standard POS tags and achieves up to 80% F_1-score. We are not aware of previous work on Quechua.

	Best on	Macro-average	Macro-average on weblogs
LSTM	5/22	0.8516	0.8525
LSTM w. HREF	17/22	0.8536	0.8540

Table 2: Small, but consistent improvement for domain adaptation for English chunking

in general.

Learning curve Hard parameter sharing makes models less prone to overfitting (Søgaard and Goldberg, 2016). Since little labeled data means higher variance, this suggests that multi-task learning is more effective in scenarios where data is scarce. This, in our case, would mean that hyperlinks reduce the need for labeled data.

It is not straight-forward, however, to interpret standard learning curves that only vary the number of training data points for the target task, since the auxiliary task may easily swamp the target task. Even if we balance auxiliary and target data sets, results can still be hard to interpret: If we fix the hyper parameters, the regularization effect of the auxiliary task reduces with less data, since our network can effectively memorize the data points and thereby discriminate between the two tasks and allocate parts of the network for each task (Zhang et al., 2017).

When we balance the amount of target and auxiliary task data, and reduce the expressivity of our networks, we observe higher gains (error reductions of up to 15%) with small amounts of data ($20 \leq n \leq 100$). The optimal balance between the target and the auxiliary task seems to favor the target task. If we subsample the auxiliary data to be proportionally smaller (n a third of target data), we see greater and more robust improvements, especially for small n.

Quechua in-sample tests We train the same models on the Quechua data. We use the Wikipedia-trained, 64-dimensional Polyglot embeddings[6] and use 32-dimensional LSTM layers. We observe a 4% error reduction, which is higher than for our English in-sample test, but smaller than the improvements on the other tasks.

4 Conclusion

Readily available data (HTML mark-up) can be used to improve partial parsers for low-resource languages, as well as state-of-the-art partial parsers for English, with improvements that carry over to new, unseen domains.

Acknowledgements

Thanks to the anonymous reviewers for their comments that helped improve the paper. This research is funded by the ERC Starting Grant LOWLANDS No. 313695.

[6] polyglot.readthedocs.io

References

Jonathan Baxter. 2000. A model of inductive bias learning. *Journal of Artificial Intelligence Research* 12:149–198.

Joachim Bingel and Anders Søgaard. 2017. Identifying beneficial task relations for multi-task learning in deep neural networks. In *EACL*.

Rich Caruana. 1993. Multitask learning: a knowledge-based source of inductive bias. In *ICML*.

Massimiliano Ciaramita and Yasemin Altun. 2006. Broad-coverage sense disambiguation and information extraction with a supersense sequence tagger. In *Proc. of Proceedings of EMNLP*. Sydney, Australia, pages 594–602.

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

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9(8):1735–1780.

Sigrid Klerke, Yoav Goldberg, and Anders Søgaard. 2016. Improving sentence compression by learning to predict gaze. In *NAACL*.

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

George A. Miller, Martin Chodorow, Shari Landes, Claudia Leacock, and Robert G. Thomas. 1994. Using a semantic concordance for sense identification. In *Proceedings of the workshop on Human Language Technology*. Association for Computational Linguistics, pages 240–243.

Barbara Plank. 2016. Keystroke dynamics as signal for shallow syntactic parsing. In *COLING*.

Annette Rios. 2015. *A Basic Language Technology Toolkit for Quechua*. Ph.D. thesis, University of Zurich.

Anders Søgaard and Yoav Goldberg. 2016. Deep multitask learning with low level tasks superviser at lower layers. In *ACL*.

Valentin Spitkovsky, Daniel Jurafsky, and Hiyan Alshawi. 2010. Profiting from mark-up: Hyper-text annotations for guided parsing. In *ACL*.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. 2017. Understanding deep learning requires rethinking generalization. In *ICLR*.