SRL4ORL: Improving Opinion Role Labelling Using Multi-Task Learning With Semantic Role Labeling

Ana Marasović
Research Training Group AIPHES
Department of Computational Linguistics
Heidelberg University
marasovic@cl.uni-heidelberg.de

Anette Frank
Research Training Group AIPHES
Department of Computational Linguistics
Heidelberg University
frank@cl.uni-heidelberg.de

Abstract

For over a decade, machine learning has been used to extract opinion-holder-target structures from text to answer the question *Who expressed what kind of sentiment towards what?*. Recent neural approaches do not outperform the state-of-the-art feature-based model for Opinion Role Labelling (ORL). We suspect this is due to the scarcity of labelled training data and address this issue using different multi-task learning techniques with a related task which has substantially more data, i.e. Semantic Role Labelling (SRL). Despite difficulties of the benchmark MPQA corpus, we show that indeed the ORL model benefits from SRL knowledge.

1 Introduction and related work

Fine-Grained Opinion Analysis aims to: detect explicit opinion expressions (O) (such as *has supported* in the example (1)[1]), measure their intensity (e.g. strong), identify their holders (H), i.e. entities that express an opinion (e.g. *Mexico*), identify their targets (T), i.e. entities or propositions at which the sentiment is directed (e.g. *the OPEC cutbacks*) and classify target-dependent sentiment (e.g. positive).

(1) Traditionally, [Mexico]$_{H_1}$ [has supported]$_{O_1(pos)}$ [the OPEC cutbacks]$_{T_1}$; however, [analysts]$_{H_2}$ [agreed]$_{O_2(pos)}$ that [it]$_{T_2,H_3}$ [will now tend to support]$_{O_3(pos)}$ [the United States]$_{T_3}$, the main world consumer, and the one that would be adversely affected by a possible increase in crude prices.

As the commonly accepted benchmark corpus MPQA [1] uses span-based annotations to represent *opinion entities* (opinions, holders and targets), the task is usually approached with sequence labeling techniques and the BIO encoding scheme [2, 3, 4]. Initially were proposed pipeline models which first predict opinion expressions and then, given an opinion, label its *opinion roles*, i.e. holders and targets [5]. Pipeline models have been substituted with the so-called joint models that simultaneously identify all opinion entities and predict which opinion role is related to which opinion [2, 3, 4]. Recently an LSTM-based joint model was proposed [4] that unlike the prior work [2, 3] does not depend on external resources (such as syntactic parsers, named entity recognizers, etc.). The neural variant does not outperform the feature-based CRF model [3] in Opinion Role Labeling (ORL).

Both the neural and the CRF joint models achieve circa 55% F1 score for predicting which targets relate to which opinions in MPQA, meaning that these models are not ready to answer the question this line of research is usually motivated with: *Who expressed what kind of sentiment towards what?*

Our goal is to investigate the limitations of neural models in solving different subtasks of fine-grained opinion analysis on MPQA and to gain a better understanding of what is solved and what is next.

[1]The example drawn from MPQA [1]. Opinions in MPQA can also be beliefs, emotions, speculations, etc.

31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA.

We suspect that one of the fundamental obstacles for neural models trained on MPQA is its small size. One way to cope with scarcity of labeled data is to use multi-task learning with appropriate auxiliary tasks. A good auxiliary task candidate for ORL is Semantic Role Labeling (SRL), the task of predicting predicate-argument structure of a sentence, which answers the question *Who did what to whom, where and when?*. For the first 23 tokens of example (1) the output of the SRL demo[2] is:

	Traditionally	,	Mexico	has	supported	the	OPEC	cutbacks	;	however	,	analysts	agreed	that	it	will	now	tend	to	support	the	United	States
support.01	AM-TMP	-	A0	-	-	A1	A1	A1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
cutback.01	-	-	-	-	-	-	A0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
agree.01	-	-	-	-	-	-	-	-	-	AM-DIS	-	A0	-	A1	A1	A1	A1	A1	A1	A1	A1	A1	A1
tend.02	-	-	-	-	-	-	-	-	-	-	-	-	-	-	A1	AM-MOD	AM-TMP	-	A2	A2	A2	A2	A2
support.01	-	-	-	-	-	-	-	-	-	-	-	-	-	-	A0	-	-	-	-	-	A1	A1	A1

If we compare the semantic roles of the predicates *support*, *agree* and *(tend to) support* we can notice significant overlap with the opinion roles of the corresponding opinions according to MPQA (marked blue). For this reason, the output of a SRL system has been commonly used for feature-based models for fine-grained opinion analysis [5, 2, 3]. Additionally, there is a considerable amount of available annotated data for training SRL models (Table 1), which made neural SRL models successful [6, 7].

Although SRL is a reasonable auxiliary task, an obstacle for properly exploiting SRL training data with MTL could be imprecisions and incompleteness of the MPQA annotations. In example (1), *agreed* is considered an opinion with positive sentiment and its target is marked as the token *it*, meaning that annotators understood that the *analysts* are positive towards *Mexico*, just because *analysts* agree that *Mexico will now tend to support the United States*. Even if they have expressed an opinion, the target would not be *it*, but *it will now tend to support the United States*. Regarding incompleteness, prior work [4] has shown that their model makes reasonable predictions in sentences which do not have annotations at all, e.g. $[\text{mothers}]_{H_1}$ $[\text{care}]_{O_1}$ for $[\text{their young}]_{T_1}$, in: *From the fact that mothers care for their young, we can not deduce that they ought to do so, Hume argued.* Both types of shortcomings in the annotation ground truth will be revealed by correct SRL annotations.

In spite of these challenges, we adopt one of the recent successful architectures for SRL [6], experiment with different multi-task learning frameworks and show that indeed an ORL model can benefit from MTL with SRL.

2 Neural MTL for SRL and ORL

As a general neural architecture for single- and multi-task learning we use the recently proposed SRL model [6] (Z&H) which successfully labels semantic roles without any syntactic guidance. This model consists of a stack of bi-directional LSTMs and a CRF which makes the final prediction. Every sentence is processed as many times as there are predicates in it. The inputs to the first LSTM are not only token embeddings but three additional features: embedding of the predicate, embedding of the context of the predicate and an indicator feature (1 if the current token is in the predicate context, 0 otherwise). Adapting this model for labeling of opinion roles is straightforward, the only difference being that opinion expressions can be multi-words and only two opinion roles are assigned: H and T.

Multi-task learning (MTL) techniques aim to learn several tasks jointly by leveraging knowledge from all tasks. In the context of neural networks, MTL is commonly used such that is predefined which layers have tied parameters and which are task-specific (i.e. hard-parameter sharing). There are various ways of defining which parameters should be shared and how to train them.

Fully-shared (FS) MTL model. In a fully-shared model (Fig. 1), all parameters of the general model except the output layer are shared. Each task has a task-specific output layer which makes the prediction based on the representation produced by the final LSTM. When training on a mini-batch of a certain task, parameters of the output layer of the other tasks are not updated.

Hierarchical MTL (H-MTL) model. For NLP applications, often some given task (high-level task) is supposed to benefit from another task (low-level task) more than other way around, e.g. parsing from POS tagging. This intuition lead to designing hierarchical MTL models [8, 9] in which predictions for low-level tasks are not made on the basis of the representation produced at the final LSTM, but on the representation produced by a lower-layer LSTM (Fig. 2).

Shared-private (SP) MTL model. In the state-private model in addition to the stack of shared LSTMs, each task has a stack of task-specific LSTMs [10] (Fig. 3). Representations at the outermost

[2] `http://barbar.cs.lth.se:8081`

	task	train size	dev size	test size	$\|\mathcal{Y}\|$	OOV rate %	$H_{\mathcal{Y}}$
CoNLL'05	SRL	90750	3248	6071	106	14.53	1.87
MPQA	ORL	3951	1498	450	7	5.97	1.03

Table 1: Datasets w/ nb. of SRL predicates/ORL opinions in train, dev & test set, size of label inventory, percent. of training words not in pre-trained GloVe embeddings, entropy of label distribution.

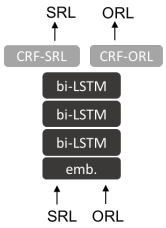

Figure 1: Fully-shared (FS) MTL model.

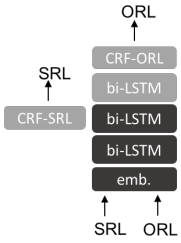

Figure 2: Hierarchical MTL model (H-MTL).

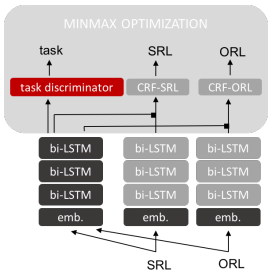

Figure 3: (Adversarial) state-private ((A)SP) MTL model.

shared LSTM and the task-specific LSTM are concatenated and passed to the task-specific output layer. This architecture enables the model to selectively utilize the shared and task-specific information.

Adversarial shared-private (ASP) model. The limitation of the SP model is that it does not prevent the shared layers from capturing task-specific features. To prune shared layers, the SP model is extended with a *task discriminator* [10]. The task discriminator (Fig. 3, marked red) predicts to which task the current batch of data belongs, based on the representation produced by the shared LSTMs. If the shared LSTMs are task-invariant, this discriminator should perform badly, so we update the shared parameters such that the entropy of the predicted task distribution is maximized. At the same time we want the discriminator to challenge the shared LSTMs, so we update the discriminator's parameters to minimize its cross-entropy loss. This minmax optimization is known as *adversarial training* and recently it gained a lot of attention for NLP applications [10, 11, 12, 13, 14, 15, 16, 17, 18].

3 Experimental setup

3.1 Datasets

For SRL we use the newswire CoNLL-2005 shared task dataset [19], annotated with PropBank predicate-argument structures. Sections 2-21 of the WSJ [20] are used for training and section 24 for devset. The test set consists of Section 23 of WSJ and 3 sections of the Brown corpus.

For ORL we use the MPQA corpus [1] which contains news documents manually annotated for opinions and other private states. We follow prior work which set aside 132 documents for development and used the remaining 350 documents for evaluation. MPQA allows annotating implicit opinions as explicit with a special implicit tag label, which in some cases annotators missed to indicate. To capture all implicit opinions, we discard one-character long opinions and opinions whose holder is the writer of the document. More data statistics can be found in Table 1.

3.2 Evaluation metrics

For both tasks we adopt evaluation metrics from prior work. For SRL, precision is defined as the proportion of semantic roles predicted by a system which are correct, recall is the proportion of gold roles which are predicted by a system, F1 score is the harmonic mean of precision and recall.

In case of ORL, we report 10-fold CV[3] with two measures: *binary F1 score* and *proportional F1 score*, for holders and targets separately. Binary F1 score is the same as the SRL F1 score described above, just for opinion roles. *Proportional recall* measures proportion of the overlap between a

[3]We used the same splits as the prior work [4]. We thank the authors for providing the information.

gold holder (target) and an overlapping predicted holder (target), *proportional precision* measures proportion of the overlap between a predicted holder (target) and an overlapping gold holder (target). Again, F1 score is the harmonic mean of the corresponding precision and recall.

3.3 Training details

Input representation. We used 100d GloVe word embeddings [21] pre-trained on Gigaword and Wikipedia and to avoid overfitting, did not fine-tune them. For MTL models vocabulary was built from all the words in the train data of both tasks, and OOV words were replaced with an UNK token. The embedding of the context a predicate or an opinion is the average of the embeddings of the predicate or the opinion phrase, of 2 preceding words and 2 words after.

Weights initialization. The size of all LSTM hidden states was set to 100. The number of the backward and the forward LSTM layers is set to 3, which counts for 6 LSTM layers in Z&H. Z&H achieved circa 2% higher SRL F1 score with 8 LSTM layers, but that deep models would cause overfitting on small-sized ORL data. In H-MTL model SRL is supervised at the 2nd LSTM layer. We initialized the LSTM weights with random orthogonal matrices [22], all other weight matrices with the *He initialization* [23]. LSTM forget biases were initialized with 1s [24], all other biases with 0s.

Optimization. We trained our model in mini-batches of size 32 using Adam [25] with the learning rate of 10^{-3}. For MTL we alternate batches from different tasks. We clip gradients by global norm [26], with a clipping value set to 10. Single-task models were trained for 10K iterations and MTL models for 20K. The entropy of the predicted task distribution is scaled with 0.05. We stop training if the arithmetic mean of proportional F1 scores of holders and targets is not improved in 3 epochs.

Regularization. We used l2-regularization on output layers with λ set to $8.35 \cdot 10^{-3}$. Dropout [27] with a keep probability $k_p \in 0.85$ was applied to the outputs of the LSTMs and to the input embeddings with $k_p \in 0.75$. HPs were not tuned.

4 Results

All models are evaluated every 1000 iterations on the ORL devset and saved if they achieve a higher arithmetic mean of proportional F1 scores of holders and targets on the ORL devset. The saved models are used for evaluation on the ORL test set. The mean of F1 scores over 10 folds, μ, and the standard deviation, σ, of all models are reported in Table 2. Evaluation metrics follow Section 3.2. We define the F1 score interval as $[\mu - \sigma, \mu + \sigma]$. The lower part of Table 2 compares the F1 scores of MTL models with the F1 scores of the single-task model (Z&H) and measures the difference between the point $\mu - \sigma$ of MTL models (P1) and the point $\mu + \sigma$ of the single-task Z&H model (P2). If P1 is larger than P2 we can expect that the difference of the F1 scores is significant.

Single-task vs. MTL. Table 2 shows that all MTL models improve in large margins over the single-task Z&H model with all evaluation measures, for both holders and targets. On devset, the F1 score intervals of FS and H-MTL do not overlap with the F1 score interval of Z&H, meaning that improvements on the dev set are probably significant. However, the picture changes on the test set. This shift in results could be due the small-size of the test set (Table 1), which results with a high-variance estimate. Indeed, if we compare standard deviations on the test set we notice they are always much higher than on devset. Larger improvements are visible in labelling of holders, which is not surprising given that holders are usually short, less ambiguous and clearly resemble the A0 role.

Comparing MTL models. Simpler models (FS, H-MTL) with 677828 parameters achieve better results than their more complex competitors (SP, ASP) with 1987630 parameters. We experimented with lowering the number of shared layers and the number of task-specific layers in the SP model to lower the number of parameters, but this did not lead to improvements. If we compare FS with H-MTL we can notice that in 2 cases H-MTL performs better, in 2 cases FS, otherwise they perform nearly the same (difference is smaller than 0.1). If we compare their F1 score intervals, they always overlap. Therefore we concluded that for ORL there is no benefit in choosing FS over H-MTL and vice versa. Finally, as expected SP benefits from adversarial training for predicting targets in devset.

SRL results. SRL results with MTL (test F1-scores in range $[60.70, 61.10]$) are lower than the performance of the single-task SRL model (test F1-score 77.51). However, this is expected as we stop training MTL models after 20K iterations, when models converge on ORL data, but not yet on

	dev (MPQA)				test (MPQA)			
	holder		target		holder		target	
	bin. F1 $\mu \pm \sigma$	prop. F1 $\mu \pm \sigma$	bin. F1 $\mu \pm \sigma$	prop. F1 $\mu \pm \sigma$	bin. F1 $\mu \pm \sigma$	prop. F1 $\mu \pm \sigma$	bin. F1 $\mu \pm \sigma$	prop. F1 $\mu \pm \sigma$
Z&H	70.61 ± 1.05	67.75 ± 1.16	69.39 ± 1.01	63.70 ± 1.47	70.63 ± 2.67	68.63 ± 2.53	70.24 ± 3.29	63.34 ± 2.45
FS	**75.39 ± 0.65**	**72.77 ± 0.63**	**73.39 ± 1.35**	67.71 ± 1.69	75.03 ± 3.42	**72.90 ± 3.08**	**75.16 ± 2.19**	68.74 ± 3.00
H-MTL	75.13 ± 0.97	72.73 ± 1.14	72.95 ± 0.72	**67.92 ± 1.02**	**75.10 ± 2.91**	72.89 ± 2.76	75.07 ± 1.55	**69.09 ± 2.21**
SP	74.21 ± 1.60	71.26 ± 1.84	71.35 ± 1.01	64.55 ± 0.80	74.43 ± 3.26	72.22 ± 3.34	73.10 ± 1.11	65.05 ± 1.83
ASP	73.88 ± 1.49	71.07 ± 1.79	71.78 ± 1.03	64.99 ± 0.99	73.57 ± 2.98	71.18 ± 2.88	73.07 ± 1.94	65.21 ± 2.59

	dev (MPQA)				test (MPQA)			
	holder		target		holder		target	
	bin. F1	prop. F1	bin. F1	prop. F1	bin. F1	prop. F1	bin. F1	prop. F1
ΔF1(FS , Z&H)	4.78	5.03	3.99	4.01	4.40	4.27	4.91	5.40
ΔF1(H-MTL , Z&H)	4.52	4.98	3.56	4.22	4.46	4.26	4.83	5.76
ΔF1(SP , Z&H)	3.60	3.51	1.96	0.85	3.80	3.58	2.86	1.71
ΔF1(ASP , Z&H)	3.27	3.32	2.39	1.29	2.94	2.54	2.82	1.88
Δ(FS $\mu - \sigma$, Z&H $\mu + \sigma$)	3.08	3.24	1.64	0.85	-1.69	-1.34	-0.56	-0.06
Δ(H-MTL $\mu - \sigma$, Z&H $\mu + \sigma$)	2.50	2.67	1.83	1.73	-1.11	-1.03	-0.01	1.09
Δ(SP $\mu - \sigma$, Z&H $\mu + \sigma$)	0.96	0.51	-0.06	-1.42	-2.13	-2.28	-1.54	-2.58
Δ(ASP $\mu - \sigma$, Z&H $\mu + \sigma$)	0.73	0.37	0.35	-1.16	-2.71	-2.87	-2.40	-3.17

Table 2: ORL results and comparison of MTL models with the single-task ORL model (Z&H).

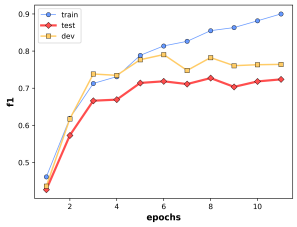

Figure 4: Learning curve of FS-MTL for holders with prop. F1.

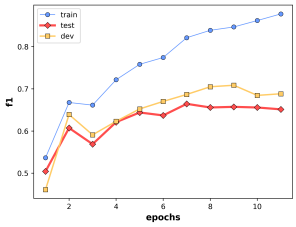

Figure 5: Learning curve of FS-MTL for targets with prop. F1.

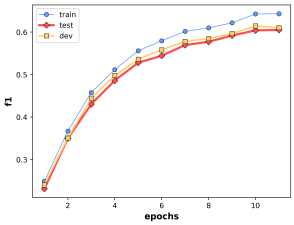

Figure 6: Learning curve of FS-MTL for SRL.

SRL data as it is visible on Figures 4–6 (1 epoch counts for 1K iterations). These figures illustrate the insight shown by the related work [28]: MTL works when the main task (ORL) has a flattening learning curve (Figures 4 & 5), but the auxiliary task (SRL) curve is still steep (Figure 6).

5 Conclusions and future directions

We address the problem of scarcity of annotated training data for labelling of opinion holders and targets (ORL) with multi-task learning (MTL) with Semantic Role Labelling (SRL). We experimented with different MTL frameworks and found that simpler MTL models achieve the best improvements over the single-task model. However, we still do not know what kind of SRL knowledge is transferred and what it is that makes simpler MTL models more successful.

Although simplicity is desirable, we expect more from MTL models; namely, *interpretability, flexibility* and the possibility of *continual learning*. Interpretability could help understanding what kind of SRL knowledge is helpful for ORL. By flexibility we mean possibilities to integrate other tasks (e.g. dependency parsing), another annotation schema (FrameNet) and cross-lingual labelling. Although this might seem trivial, the obstacle is that the different (NLP) tasks perform best with different types of architectures. The SP model can be extended with different architectures for different tasks, languages and annotation schemas. Finally, we would like to learn from all tasks beneficial for ORL (over different languages, datasets and annotation schemes), never forget gained knowledge and let the model decide what to use. Shared layers in the SP model can be seen as off-the-shelf knowledge and be used for unseen new tasks [10].

In future work we will design a model that marries the simplicity of vanilla MTL models with the flexibility, possibility of continual learning and interpretability of the more complex MTL models.

Acknowledgments

This work has been supported by the German Research Foundation as part of the Research Training Group Adaptive Preparation of Information from Heterogeneous Sources (AIPHES) under grant No. GRK 1994/1. We would like to thank anonymous reviewers for useful comments.

References

[1] Janyce Wiebe, Theresa Wilson, and Claire Cardie. Annotating Expressions of Opinions and Emotions in Language. *Language Resources and Evaluation*, 39:165–210, 2005.

[2] Yejin Choi, Eric Breck, and Claire Cardie. Joint extraction of entities and relations for opinion recognition. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 431–439, Sydney, Australia, July 2006. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/W/W06/W06-1651`.

[3] Bishan Yang and Claire Cardie. Joint Inference for Fine-grained Opinion Extraction. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1640–1649, Sofia, Bulgaria, August 2013. URL `http://www.aclweb.org/anthology/P13-1161`.

[4] Arzoo Katiyar and Claire Cardie. Investigating LSTMs for Joint Extraction of Opinion Entities and Relations. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 919–929, Berlin, Germany, August 2016. URL `http://www.aclweb.org/anthology/P16-1087`.

[5] Soo-Min Kim and Eduard Hovy. Extracting Opinions, Opinion Holders, and Topics Expressed in Online News Media Text. In *Proceedings of the Workshop on Sentiment and Subjectivity in Text*, pages 1–8, Sydney, Australia, July 2006. URL `http://www.aclweb.org/anthology/W/W06/W06-0301`.

[6] Jie Zhou and Wei Xu. End-to-end learning of semantic role labeling using recurrent neural networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1127–1137, Beijing, China, July 2015. URL `http://www.aclweb.org/anthology/P15-1109`.

[7] Bishan Yang and Tom Mitchell. A joint sequential and relational model for frame-semantic parsing. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1258–1267, Copenhagen, Denmark, September 2017. URL `https://www.aclweb.org/anthology/D17-1129`.

[8] Anders Søgaard and Yoav Goldberg. Deep multi-task learning with low level tasks supervised at lower layers. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 231–235, Berlin, Germany, August 2016. Association for Computational Linguistics. URL `http://anthology.aclweb.org/P16-2038`.

[9] Kazuma Hashimoto, caiming xiong, Yoshimasa Tsuruoka, and Richard Socher. A joint many-task model: Growing a neural network for multiple nlp tasks. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 446–456, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. URL `https://www.aclweb.org/anthology/D17-1046`.

[10] Pengfei Liu, Xipeng Qiu, and Xuanjing Huang. Adversarial multi-task learning for text classification. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1–10, Vancouver, Canada, July 2017. Association for Computational Linguistics. URL `http://aclweb.org/anthology/P17-1001`.

[11] Xinchi Chen, Zhan Shi, Xipeng Qiu, and Xuanjing Huang. Adversarial multi-criteria learning for chinese word segmentation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1193–1203, Vancouver, Canada, July 2017. Association for Computational Linguistics. URL `http://aclweb.org/anthology/P17-1110`.

[12] Young-Bum Kim, Karl Stratos, and Dongchan Kim. Adversarial adaptation of synthetic or stale data. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1297–1307, Vancouver, Canada, July 2017. Association for Computational Linguistics. URL `http://aclweb.org/anthology/P17-1119`.

[13] Lianhui Qin, Zhisong Zhang, Hai Zhao, Zhiting Hu, and Eric Xing. Adversarial connective-exploiting networks for implicit discourse relation classification. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1006–1017, Vancouver, Canada, July 2017. Association for Computational Linguistics. URL `http://aclweb.org/anthology/P17-1093`.

[14] Yi Wu, David Bamman, and Stuart Russell. Adversarial training for relation extraction. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1779–1784, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/D17-1187.

[15] Tao Gui, Qi Zhang, Haoran Huang, Minlong Peng, and Xuanjing Huang. Part-of-speech tagging for twitter with adversarial neural networks. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2401–2410, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/D17-1255.

[16] Jiwei Li, Will Monroe, Tianlin Shi, Sébastien Jean, Alan Ritter, and Dan Jurafsky. Adversarial learning for neural dialogue generation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2147–2159, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/D17-1229.

[17] Meng Zhang, Yang Liu, Huanbo Luan, and Maosong Sun. Adversarial training for unsupervised bilingual lexicon induction. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1959–1970, Vancouver, Canada, July 2017. Association for Computational Linguistics. URL http://aclweb.org/anthology/P17-1179.

[18] Shafiq Joty, Preslav Nakov, Lluís Màrquez, and Israa Jaradat. Cross-language learning with adversarial neural networks. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 226–237, Vancouver, Canada, August 2017. Association for Computational Linguistics. URL http://aclweb.org/anthology/K17-1024.

[19] Xavier Carreras and Lluís Màrquez. Introduction to the CoNLL-2005 shared task: Semantic role labeling. In *Proceedings of the Ninth Conference on Computational Natural Language Learning (CoNLL-2005)*, pages 152–164, Ann Arbor, Michigan, June 2005. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/W/W05/W05-0620.

[20] Eugene Charniak, Don Blaheta, Niyu Ge, Keith Hall, John Hale, and Mark Johnson. Bllip 1987-89 wsj corpus release 1. *Linguistic Data Consortium, Philadelphia*, 36, 2000.

[21] Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar, October 2014. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/D14-1162.

[22] Mikael Henaff, Arthur Szlam, and Yann LeCun. Recurrent orthogonal networks and long-memory tasks. In *Proceedings of the 33rd International Conference on Machine Learning (ICML)*, pages 2034–2042, 2016.

[23] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015.

[24] Rafal Jozefowicz, Wojciech Zaremba, and Ilya Sutskever. An empirical exploration of recurrent network architectures. In *Proceedings of the 32nd International Conference on Machine Learning (ICML)*, pages 2342–2350, 2015.

[25] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *Proceedings of the International Conference on Learning Representations (ICLR)*, San Diego, 2015.

[26] Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. On the difficulty of training recurrent neural networks. In *Proceedings of the 31st International Conference on Machine Learning (ICML)*, pages 1310–1318, 2013.

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

[28] Joachim Bingel and Anders Søgaard. Identifying beneficial task relations for multi-task learning in deep neural networks. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 164–169, Valencia, Spain, April 2017. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/E17-2026.