ProTestA: Identifying and Extracting Protest Events in News
Notebook for ProtestNews Lab at CLEF 2019

Angelo Basile[1] and Tommaso Caselli[2]

[1] Symanto Research GmbH & Co., Nürnberg, Germany
angelo.basile@symanto.net
www.symanto.net
[2] Rijksuniversiteit Groningen, Groningen, the Netherlands
t.caselli@rug.nl

Abstract. This notebook describes our participation to the Protest-New Lab, identifying protest events in news articles in English. Systems are challenged to perform unsupervised domain adaptation against three sub-tasks: document classification, sentence classification, and event extraction. We describe the final submitted systems for all sub-tasks, as well as a series of negative results. Results indicate pretty robust performances in all tasks (average F1 of 0.705 for the document classification sub-task, average F1 of 0.592 for the sentence classification sub-task; average F1 0.528 for the event extraction sub-task), ranking in the top 4 systems, although drops in the out-of-domain test sets are not minimal.

Keywords: document classification · sentence classification · event extraction · protest events

1 Introduction

The growth of the Web has made more and more data available, and the need for Natural Language Processing (NLP) systems that are able to generalize across data distributions has become a urgent topic. In addition to this, portability of models across data sets, even when assumed to be on the same domain, is still a big challenge in NLP. Indeed recent studies have shown that systems, even when using architectures based on Neural Networks and distributed word representations, are highly dependent on their training sets and can hardly generalise [12,30,5].

The 2019 CLEF ProtestNews Lab [3] targets models' portability and unsupervised domain adaptation in the area of social protest events to support comparative social studies. The lab is organised along three tasks: *a.)* document

[3] https://emw.ku.edu.tr/clef-protestnews-2019/

classification (Task 1); *b.*) sentence classification (Task 2); and *c.*) event trigger and argument extraction (Task 3).

Task 1 and 2 are essentially text classification tasks. The goal is to distinguish between documents and sentences that report on or contain mentions of protest events. Task 3 is an event extraction task, where systems have to identify the correct event trigger, in this case a protest event, and its associated arguments in every sentence of a document.

As described in [7], the creation of the data sets followed a very detailed procedure to ensure maximal agreement among the annotators as well as to avoid errors. Furthermore, the task is designed as a cascade of sub-tasks: first, identify if a document reports a protest event (Task 1), then identify which sentences are actually describing the protest event in the specific document (Task 2), and, finally, for each protest event sentence, identify the actual event mention(s) and its arguments (Task 3). However, there is no overlap among the training and test data of the three tasks.

As already mentioned, the lab's main challenge is unsupervised domain adaptation. The lab organisers made available training and development data for one domain, namely news reporting protest events in India, and asked the participants to test their models both on in-domain data and on out-of-domain ones, namely news about protest events in China. In the remainder of the notebook, we will refer to these two test distributions as India and China.

When analysing the three tasks, it seems evident that the first two tasks are very similar and can be targeted with a common architecture, and possibly features, modulo the granularity of the text message, i.e. full document *vs.* sentences. On the other hand, the third task requires a dedicated and radically different approach.

In the remainder of this contribution, we illustrate the systems we developed for the final submissions, and provide some data analysis that may help understand the drops in performance across the two test data. We also describe and reflect on what we tried but did not work as expected.

2 Final Systems

The three tasks have been addressed with two separate systems. In particular, for Task 1 and 2, we opted for a feature based stacked ensemble model based on a set of different basic Logistic Regression classifiers, while for Task 3, we used a Bi-LSTM architecture optimized for sequence labelling task.

2.1 Training Materials

The lab organisers made available training and development data for each task. Table 1, summarises the distributions of the labels of the training and development data for Task 1 and 2, i.e. document and sentence classification, respectively. As the figures show, the positive class, i.e. the protest documents or sentences, is pretty much unbalanced with respect to the negative one, i.e.

non-protest, ranging between 22.41% for Task 1 to 16.78% in Task 2 in training. The distribution of the classes is mirrored in the development data, with minor differences for Task 2, where the positive class is slightly bigger than the negative one (20.81% *vs.* 16.78%). For training our systems, we did not use any additional training material. The development data was used to identify which methods to use for the final systems rather than fine tuning the models, given the fact that at test time the models has to perform optimally for two different data distributions, in- and out-of-domain (India *vs.* China).

Table 1. Distributions of classes for training and development for Task 1 and Task 2. Numbers in parentheses indicate percentages.

Task	**Data set**	**Protest**	**Not Protest**
Task 1 (Document Classification)	Train	769 (22.41%)	2,661 (77.58%)
	Dev.	102 (22.31%)	355 (77.68%)
Task 2 (Sentence Classification)	Train	988 (16.78%)	4,897 (83.21%)
	Dev.	138 (20.81%)	525 (79.18%)

Table 2 illustrates the distribution of the annotations for Task 3, i.e. event trigger and argument detection. The data was released in the form of tab-delimited files, with two columns only: the first with pre-tokenized tokens and the second with labels for both event triggers and arguments. Overall, seven different argument types were annotated, namely *participant*, *organiser*, *target*, *etime* (event time), *place*, *fname* (facility name), and *loc* (location). The role set is inspired by the Automatic Content Extraction (ACE) guidelines for events [4], and especially the event types Attack and Demonstrate, although the organisers have a finer granularity for locations, as well as for people, or entities, involved in a protest distinguishing, for instance, between organisers, targets, and actual participants. The annotation are encoded in a BIO scheme (Beginning, Inside, Outside), resulting in different alphabets for event triggers (e.g. B-*trigger*, I-*trigger* and O) and each of the arguments (e.g. O, B-*organiser*, I-*organiser*, B-*etime*, I-*etime*, etc.).

Table 2. Distribution of event triggers and arguments for Task 3.

Annotations	**Train**	**Dev**
Event Triggers	844	126
Arguments	1,895	288

The training data contains 250 documents and a total of 594 sentences, while the development set is composed by 36 documents for a total of 171 sentences.

[4] https://www.ldc.upenn.edu/sites/www.ldc.upenn.edu/files/english-events-guidelines-v5.4.3.pdf

The average amount of event trigger per sentence is 1.41 in training and 1.35 in development, indicating that multiple event triggers are available in the same sentence. As for the arguments, the average per event trigger is 2.24 in training and 1.85 in development, indicating both that arguments are shared among event triggers in the same sentences and that not all arguments are available in every sentence. Similarly to Task 1 and 2, only the available training data was used to train the system.

2.2 Classifying Documents and Sentences (Task 1 and Task2)

The document and sentence classification tasks have been formulated as standard classification tasks. In the perspective of maximizing the system results on both test distributions, we have developed a stacked ensemble model of Logistic Regression classifiers, following a previously successful implementation that showed robust portability [18].

We extracted three different sets of features. Each set of features was used to train a basic Logistic Regression classifier, as available in the scikit learn platform [20], and obtain a 10-fold cross-validation prediction for each training set. This means that for each document/sentence, we have 3 basic classifiers as well as their corresponding predictions, resulting in a total of 6 meta level features (3 classifiers X 2 classes per each task) per document/sentence as input for the final meta-classifier. The meta-classifier is an additional Logistic Regression classifier. In training, we have used the default value of the C parameter and balanced class weights. Pre-processing of the data is limited to lowercase, and removal of special characters (e.g. #, $*$, (, . . .) and digits.

Word embeddings features We used the pre-trained FastText embedding [3] for English with 300 dimensions and sub-word information. [5] For each document/sentence, we obtain a 300 dimension representation by applying average pooling on the token embeddings, any time that a token is not present in the embedding vocabulary, we extract sub-words of length 3 or greater and check if they are present in the embeddings. This is a strategy to maximize the information in the training data as well as to reduce out of vocabulary (OOV) tokens across the different test distributions.

Most important token and character n-grams features These two sets of features have been identified as useful features to increase the robusteness and portability of the models across data sets. The features have been extracted by performing two sets of TF-IDF scores over each training data (i.e. Task 1 and Task 2) to select the most important tokens and characters n-gram per class (i.e. protest documents/sentences *vs.* non-protest documents/sentences). For each extracted token, the maximum and minimum cosine similarity is obtained with respect to all tokens in a document/sentence using the FastText embeddings. Similarly to

[5] We used the `wiki-news-300d-1M-subword.vec` model, available at `https://fasttext.cc/docs/en/english-vectors.html`.

the word embeddings feature, in case a token is not present in the embeddings, we checked for sub-words embeddings. The character n-grams are represented by means of Boolean features indicating whether they are present or not in the document/sentence, thus capturing and representing different information.

Table 3 illustrates the settings used to tune the system for each test distributions and task. The amounts of token and character n-grams varies per task as well as per test distribution. Although more experiments are needed, during the submission phase and quite not surprisingly, we observed that the higher the number of tokens and character n-grams is extracted, the better the model performs on the same test data distribution, thus loosing in portability (for instance, with 4,000 token n-grams and 1,000 characters n-grams, the F1 of Task 2 on China drops to 0.553 to 0.536).

Table 3. Most important tokens and character n-grams features for Task 1 and Task 2 across the two test distributions, i.e. India and China.

	Feature Type	**Test Set**	**Amount**
TASK 1	Tokens	India	8,000
	Char. n-grams	India	750
	Tokens	China	4,000
	Char. n-grams	China	750
TASK 2	Tokens	India	4,000
	Char. n-grams	India	1,000
	Tokens	China	2,000
	Char. n-grams	China	500

2.3 Extracting Events and their Arguments (Task 3)

We framed the event mention and argument extraction task as a supervised sequence labelling classification problem, following a well established practice in NLP [1,8,26,2,19]. In particular, given a sentence, S, the system is asked to identify all linguistic expressions $w \in S$, where w is a mention of a protest event, ev_w, as well as all linguistic expressions $y \in S$ where y is a mention of an argument, arg_y, associated to a specific event mention ev_w.

We have implemented a two-step approach using a common sequence labelling model based on a publicly available Bi-LSTM network with a CRF classifier as last layer [25]. [6] In more details, we developed two different models: first, we detect event trigger mentions, and subsequently, the event arguments (and their roles). Such an approach is inspired by SRL architectures [27], where first predicates are identified and disambiguated, and afterwards the relevant arguments are labelled. In our case, we first identify sentences that contain relevant event triggers, and then look for event arguments in these sentences.

[6] `https://github.com/UKPLab/emnlp2017-bilstm-cnn-crf`

We did not fine-tuned the hyperparameters, but followed the suggestions in [25,24] for sequence labelling tasks. In Table 4, we report the shared parameters of the networks for both tasks. The "LSTM Layers" refers separately to the number of forward and backward layers.

Table 4. System details

Parameters	**Value**
LSTM Layers	1
Units per layer	100
Optimizer	Nadam
Gradient Normalisation	$\tau = 1$
Dropout	Variational, (0.5, 0.5)
Batch size	12

Training is stopped after 5 consecutive epochs with no improvements. Komninos and Manandhar ([11]) pre-trained word embeddings are used to initialize the ELMo embeddings [22] and fine-tune them with respect to the training data. The ELMo embeddings are used to enhance the network generalisation capabilities for event and argument detection over both test data distributions. As for the event trigger detection sub-task, the embedding representations are further concatenated with character-level embeddings [15], and parts-of-speech (POS) embeddings. POS tags have been obtained from the Stanford CoreNLP toolkit [16]. [7] This minimal set of features is further extended with embedding representations for dependency relations and event triggers for the argument detection sub-task. At test time, the protest event triggers are obtained from the event mentions model.

Both for the event trigger and argument detection sub-tasks, we have conducted five different runs to better asses the variability of the deep learning models due to random initialisations. At test time, we selected the model that obtained the best F1 scores on the development set out of the five runs.

3 Results on Test Data

In this section we illustrate the results for all three tasks [8]. Notice that for Task 3 the scores are cumulative for both event trigger and participant detection. For all tasks, the ranking is based on the average F1 of the systems on the two test distributions (i.e. India and China). Table 5 reports the results for each task and the corresponding rankings. For ease of comparison, we also report the distance from the best ranking system for each task expressed in differences in F1 scores.

[7] We used version 3.9.2

[8] Results and ranking were taken from the Codalab page of the ProtestNews Lab available at `https://competitions.codalab.org/competitions/22349#results` .

Table 5. Results and ranking for Task 1, 2, and 3.

Data set	Task	F1 score	Avg. F1	Final Rank	-1st
India	Task 1	0.807	0.702	3	-0.044
China	Task 1	0.597			
India	Task 2	0.631	0.592	4	-0.062
China	Task 2	0.553			
India	Task 3	0.600	0.528	3	-0.039
China	Task 3	0.456			

Quite disappointingly, the drops against the China test data are not minimal, although with a pretty wide range. The minimal drop is on Task 2, where the system does not perform optimally on the India test data (F1 0.631). On the other hand, the largest drop is in Task 1, where the system looses 0.21 points in F1 when applied to the China data set. As for Task 3, the drop is still relevant (-0.144 points). However, we also noticed that the results for Precision and Recall on the China are pretty well balanced (P = 0.492; R = 0.425), indicating some robustness of the trained model.

4 Data Analysis

To better understand the results of our systems for the three tasks, we have conducted an analysis of the data to highlight similarities and differences. Following recent work [23,13], we embrace the vision that corpora, even when on the same domain, are not monolithic entities but rather they are regions in a high dimensional space of latent factors, including topics, genres, writing styles, years of publication, among others, that express similarities and diversities. In particular, we have investigated to what extent the test sets of the three tasks occupy a similar (or different) portions of this space with respect to their corresponding training distributions. To do so, we used two metrics, the Jensen-Shannon (J-S) divergence and the out-of-vocabulary rate of tokens (OOV), that previous work in transfer learning [28] has shown to be particularly useful for this kind of analysis.

The J-S divergence assesses similarity between two probability distributions, q and r and is a smoothed, symmetric variant of the Kullback-Leibler divergence. On the other hand, the OOV rate can be used to assess the differences between data distributions, as it highlights the percentage of unknown tokens. All measures have been computed between the training data and the two test distributions, i.e. India and China. Results are reported on Table 6.

As the figures in Table 6 show the test distributions, India and China, can be seen as occupying pretty different portions of this variety of spaces. Not surprisingly, the China distributions are very different from their respective training ones, although with varying degrees. This variation in similarity, however, also affects the India test distributions, where the highest similarity is observed for Task 1 (0.922) and the lowest for Task 3 (0.703). The J-S scores show that the

Table 6. Similarity, J-S, and Diversity, OVV, between train and test distributions for all tasks

	J-S		**OOV**	
Task	**India**	**China**	**India**	**China**
Task 1 (Document Classification)	0.922	0.822	28.11%	50.25%
Task 2 (Sentence Classification)	0.822	0.743	29.41%	41.12%
Task 3 (Event Extraction)	0.703	0.575	44.33%	53.82%

test distributions for Task 3 and Task 2 are even more different than those for Task 1, indicating that the differences in performances of the models across the three tasks is subject to these variations in similarities. As a further support to this observation, we found that there is a positive significant correlation between the J-S similarity scores and the F1 values across the three tasks (ρ=0.901, p<.05).

The OOV rates support the observations conducted with the J-S divergence. The OOV rates for the India test distributions are much lower then those compared to China, clearly signalling that there are strong lexical differences among the data sets. The OOV rate for India and China for Task 3 are much closer than those for Task 1 and 2, and still the differences in overall F1 scores for this task between the two test distributions is pretty large (F1 0.600 for India *vs.* F1 0.456 for China), suggesting that OOV is actually a less powerful predictor of differences in performance between data distributions. Indeed, we have found that there is a negative non significant correlation between OOV rates and the F1 scores across the tasks (ρ=-0.804, p>.05).

Finally, a further aspect to account for the behavior of the models concerns the proportion of the predictions. In particular, for Task 1 and 2 the proportions of the predictions in the two classes (protest *vs.* non-protest) are the same as in the training sets for the India data (i.e. in-domain), while they drop when applied to the China tests. In Task 3, the system predicts the same proportion of event triggers in both test distributions (0.90 event per sentence on India, and 0.95 event per sentence on China, respectively), although slightly lower than that observed in training. On the other hand, the proportions of predicted arguments per event are different: on the India data, they are in line with those of the training sets (2.51 *vs.* 2.24 in training, respectively), while they are lower in the China data (1.94 *vs.* 2.24 in training, respectively). These observations further indicate that the systems for Tasks 1 and 2 are more dependent on the training set, while the system for Task 3 appears to be more resilient to out-of-domain data.

5 Alternative Methods: What Did Not Work

In this section we briefly report on alternative methods that actually resulted to be detrimental for the performance with respect to the final settings. We mainly focused on changing strategies in modelling by using different algorithms and paradigms rather than attempting to extend the training materials.

Task 1 and 2 - Inductive Transfer Learning In an attempt to build a system that better generalizes across data sets, we tried exploiting recent advancements in transfer learning. Combining a fine-tuned language model with a classifier has been shown to be a sound strategy for classifying text [6]. We thus experimented with two pre-trained contextualized embedding models, ELMo [22,21] and BERT [4]. In both cases, we extracted fixed sentence representations and used them in combination with a linear SVM with the default hyper-parameters provided by the scikit-learn implementation [20]. With BERT, we obtain fixed representations by applying average pooling to the second-to-last hidden layer using the pre-trained BERT base model. For Task 1, we represented a document as the average of the sentence embeddings obtained by using ELMo or BERT. We used spaCy for splitting the document into sentences. We experimented with frozen and fine-tuned weights. We fine-tune the inner three layers with ELMo, while we started from the pre-trained base English model and trained it for 10,000 steps on the Indian training set with BERT. In this latter case, training data was assembled by combining the document and sentence training corpora. Finally, we also experimented with an ensemble model using both word and character n-grams and BERT embedding representations.

We obtained promising results on the development set, but, surprisingly, the performances dropped when applied to the test distributions (F1 = 0.466 for ELMo, and F1 = 0.567 for BERT, respectively). The ensemble model using both dense and sparse representations outperformed the simpler model by 0.1 F1 point.

Task 1 and 2 - Convolutional Neural Networks (CNN) It has been shown that character-level convolutional neural networks perform well in document and sentence classification tasks [31] and, being character based, these models are in theory not severely harmed by OOV words, thus making portability across test distributions less prone to errors. We experimented with the architecture described in [10], randomly initializing character embeddings, which are then passed as input to a stack of convolutional networks, with kernel sizes ranging from 3 to 7. As a regularization method, we use a 10% dropout [29]. Similarly to the use of the inductive transfer leaning approaches, good results on the development set were followed by very poor test results (F1 = 0.427). As for this approach, we hypothesize that the training data is too small for effectively using randomly initialized embeddings, although character-based.

Task 1 and 2 - FastText We experimented with Facebook's FastText system, an an off-the-shelf supervised classifier [9]. We trained two versions of the system using different token n-grams representations (i.e. bigrams and trigrams), the `wiki-news-300d-1M-subword.vec` FastText embeddings with subwords for English, and varying learning rates (ranging from 0.1 up to 1.0). We fine tuned the learning rate against the development data. Pre-processing of the data is the same as the one used for the final system, namely lowercase and removal of special characters (e.g. #, $*$, (, . . .) and digits. When applied to the test data, the best model scored F1 0.608 We also observed that bigrams performs optimally for Task 1, regardless of the test distributions, while for Task 2, bigrams worked best for the in-domain test distributions, i.e. India, and trigrams for the out-of-domain one, i.e. China.

Task 3 - Multi-task Learning We investigated if a multi-task learning architecture, still based on the Bi-LSTM network, could be a viable solutions to improve the system performance and portability. Given the incompatibility of the Task 3 annotations with other existing corpora for event extraction (e.g. ACE and POLCON [14]), opted for a multi-task learning approach, as it has proven useful to address scarcity of labeled data. However, we adopted a slightly different strategy: rather than using an alternative tasks in support of our target task, e.g. semantic role labelling in support of opinion role labelling [17], we used an alternative data set annotated with different labels but targeting the same problem. We thus extracted all sentences annotated with Attack and Demonstrate events from the ACE corpus and used them as a support task in a multi-task learning setting. In this case, we achieved an averaged F1 of 0.517 for both test distributions, lower than 0.011 points than the final submitted system. On the positive side, however, we observed that the multi-task model obtains the best Precision score on the China test data (0.541), although at a large expense of Recall.

6 Conclusion and Future Work

Our contribution mainly focused on two aspects: *a.*) assess the most viable approach for each task at stake and maximize portability with limited efforts; *b.*) explain the limits of the trained models in terms of similarities and differences across training and test distributions rather than just limiting to technical aspects of the systems.

Task 1 and Task 2 have shown that a simple system can obtain competitive results in an unsupervised domain adaptation setting. This aspect is actually encouraging and triggers further investigation in this direction by focusing efforts on parameter optimisation. We also believe that the lack of any material for the out-of-domain distributions is a further challenge to take into account, as no fine tuning of the models on the target domain was actually possible. As far as we can put efforts into the development of maximally "generalisable" systems, the dependance of the models on the training materials remains high, thus posing the problem if we are not just modelling data sets rather than linguistic phenomena.

Task 3 has actually highlighted the contribution of both more complex architectures, such as a Bi-LSTM-CRF network, and contextualised embedding representations, such as ELMo. In this specific case, the trained model is able to predict a comparable amount of event triggers between the two test distributions, although it suffers on the argument sub-task, where less arguments are predicted for the out-of-domain data. Unfortunately, the evaluation format does not allow to quantify the losses per sub-task and per trained models.

Finally, the similarity and diversity measures (i.e. J-S divergence and OOV) resulted in useful tools to better understand the different behaviors of the systems on both test distributions. It is worth noticing how J-S similarity scores correlates with F1 scores of the trained models, suggesting that it could be possible to quantify, or predict, a margin loss of systems before applying them to out-of-domain test distributions and, consequently, take actions to minimize the losses.

References

1. Ahn, D.: The stages of event extraction. In: Proceedings of the Workshop on Annotating and Reasoning about Time and Events. pp. 1–8. Association for Computational Linguistics (2006)
2. Bethard, S.: ClearTK-TimeML: A minimalist approach to TempEval 2013. In: Second Joint Conference on Lexical and Computational Semantics (* SEM). vol. 2, pp. 10–14 (2013)
3. Bojanowski, P., Grave, E., Joulin, A., Mikolov, T.: Enriching word vectors with subword information. arXiv preprint arXiv:1607.04606 (2016)
4. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
5. Glockner, M., Shwartz, V., Goldberg, Y.: Breaking nli systems with sentences that require simple lexical inferences. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers). pp. 650–655. Association for Computational Linguistics (2018), `http://aclweb.org/anthology/P18-2103`
6. Howard, J., Ruder, S.: Universal language model fine-tuning for text classification. In: ACL (2018)
7. Hürriyetoğlu, A., Yörük, E., Yüret, D., Yoltar, Ç., Gürel, B., Duruşan, F., Mutlu, O.: A task set proposal for automatic protest information collection across multiple countries. In: Azzopardi, L., Stein, B., Fuhr, N., Mayr, P., Hauff, C., Hiemstra, D. (eds.) Advances in Information Retrieval. pp. 316–323. Springer International Publishing, Cham (2019)
8. Ji, H., Grishman, R.: Refining event extraction through cross-document inference. Proceedings of ACL-08: HLT pp. 254–262 (2008)
9. Joulin, A., Grave, E., Bojanowski, P., Mikolov, T.: Bag of tricks for efficient text classification. arXiv preprint arXiv:1607.01759 (2016)
10. Kim, Y.: Convolutional neural networks for sentence classification. In: EMNLP (2014)
11. Komninos, A., Manandhar, S.: Dependency based embeddings for sentence classification tasks. In: Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. pp. 1490–1500 (2016)
12. Lake, B.M., Baroni, M.: Still not systematic after all these years: On the compositional skills of sequence-to-sequence recurrent networks. arXiv preprint arXiv:1711.00350 (2017)
13. Liakata, M., Saha, S., Dobnik, S., Batchelor, C., Rebholz-Schuhmann, D.: Automatic recognition of conceptualization zones in scientific articles and two life science applications. Bioinformatics **28**(7), 991–1000 (Apr

2012). https://doi.org/10.1093/bioinformatics/bts071, http://dx.doi.org/10.1093/bioinformatics/bts071

14. Lorenzini, J., Makarov, P., Kriesi, H., Wueest, B.: Towards a dataset of automatically coded protest events from english-language newswire documents. In: Paper presented at the Amsterdam Text Analysis Conference (2016)
15. Ma, X., Hovy, E.: End-to-end sequence labeling via bi-directional lstm-cnns-crf. arXiv preprint arXiv:1603.01354 (2016)
16. Manning, C., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S., McClosky, D.: The Stanford CoreNLP Natural Language Processing Toolkit. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics. System Demonstrations. pp. 55–60 (2014)
17. Marasović, A., Frank, A.: SRL4ORL: Improving opinion role labeling using multi-task learning with semantic role labeling. In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers). pp. 583–594. Association for Computational Linguistics, New Orleans, Louisiana (Jun 2018). https://doi.org/10.18653/v1/N18-1054, https://www.aclweb.org/anthology/N18-1054
18. Montani, J.P.: Tuwienkbs at germeval 2018: German abusive tweet detection. In: 14th Conference on Natural Language Processing KONVENS 2018. p. 45 (2018)
19. Nguyen, T.H., Grishman, R.: Event detection and domain adaptation with convolutional 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 2: Short Papers). vol. 2, pp. 365–371 (2015)
20. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al.: Scikit-learn: Machine learning in python. Journal of machine learning research **12**(Oct), 2825–2830 (2011)
21. Peters, M., Ruder, S., Smith, N.A.: To tune or not to tune? adapting pretrained representations to diverse tasks. arXiv preprint arXiv:1903.05987 (2019)
22. Peters, M.E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., Zettlemoyer, L.: Deep contextualized word representations. In: Proc. of NAACL (2018)
23. Plank, B., Van Noord, G.: Effective measures of domain similarity for parsing. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1. pp. 1566–1576. Association for Computational Linguistics (2011)
24. Reimers, N., Gurevych, I.: Optimal hyperparameters for deep lstm-networks for sequence labeling tasks. CoRR **abs/1707.06799** (2017), http://arxiv.org/abs/1707.06799
25. Reimers, N., Gurevych, I.: Reporting score distributions makes a difference: Performance study of lstm-networks for sequence tagging. In: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing. pp. 338–348. Association for Computational Linguistics, Copenhagen, Denmark (September 2017), https://www.aclweb.org/anthology/D17-1035
26. Ritter, A., Etzioni, O., Clark, S., et al.: Open domain event extraction from twitter. In: Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 1104–1112. ACM (2012)
27. Roth, M., Lapata, M.: Neural semantic role labeling with dependency path embeddings. In: Proceedings of the 54th Annual Meeting of the

Association for Computational Linguistics (Volume 1: Long Papers). pp. 1192–1202. Association for Computational Linguistics, Berlin, Germany (Aug 2016). https://doi.org/10.18653/v1/P16-1113, https://www.aclweb.org/anthology/P16-1113

28. Ruder, S., Plank, B.: Learning to select data for transfer learning with bayesian optimization. In: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing. pp. 372–382. Association for Computational Linguistics, Copenhagen, Denmark (September 2017), https://www.aclweb.org/anthology/D17-1038
29. Srivastava, N., Hinton, G.E., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.R.: Dropout: a simple way to prevent neural networks from overfitting. Journal of Machine Learning Research **15**, 1929–1958 (2014)
30. Weber, N., Shekhar, L., Balasubramanian, N.: The fine line between linguistic generalization and failure in seq2seq-attention models. arXiv preprint arXiv:1805.01445 (2018)
31. Zhang, X., Zhao, J.J., LeCun, Y.: Character-level convolutional networks for text classification. In: NIPS (2015)