WUT at SemEval-2019 Task 9: Domain-Adversarial Neural Networks for Domain Adaptation in Suggestion Mining

Mateusz Klimaszewski **Piotr Andruszkiewicz**
Institute of Computer Science
Warsaw University of Technology, Poland
mk.klimaszewski@gmail.com, P.Andruszkiewicz@ii.pw.edu.pl

Abstract

We present a system for cross-domain suggestion mining, prepared for the SemEval-2019 Task 9: Suggestion Mining from Online Reviews and Forums (Subtask B). Our submitted solution for this text classification problem explores the idea of treating different suggestions' sources as one of the settings of Transfer Learning - Domain Adaptation. Our experiments show that without any labeled target domain examples during training time, we are capable of proposing a system, reaching up to 0.778 in terms of F_1 score on test dataset, based on Target Preserving Domain-Adversarial Neural Networks.

1 Introduction

Suggestion mining is an emerging task in a natural language processing (NLP) field. Definition of suggestion mining task differs in NLP's community. Close areas of study like opinion mining or sentiment analysis get a lot of attention not only from academic, but also industrial researchers. From a linguistic point of view, while these areas treat neutral polarity of a statement as an absence of opinion (Liu, 2009), suggestion does not have to be connected with positive or negative emotion and can be treated as complementary information (Negi and Buitelaar, 2015). Lack of sensitivity to statement's sentiment and various suggestions' realization strategies (Martínez Flor, 2005) make suggestion mining task interesting and challenging from a NLP's standpoint.

In this work, we present a system for cross-domain suggestion mining, ranked in the 7^{th} place in SemEval-2019 Task 9 Subtask B. The training data for this task was collected from feedback posts on Universal Windows Platform. On the other hand, the test dataset comes from the different domain of hotel reviews from the TripAdvisor website (Negi et al., 2019). In this work we will refer to those datasets' domain as *source domain* and *target domain* accordingly. For suggestion mining task in this context, we employ ensemble of Domain-Adversarial Neural Networks (DANN) where we use Structured Self-Attentive Sentence Embedding (Lin et al., 2017) as a feature extractor. Moreover, to achieve better adaptation towards target domain, we follow the approach of Gui et al. (2017) for the part-of-speech tagging and extend DANN with a Target Preserving component in a form of words decoder for target domain sentences. We train all of the parts of the described system using modified domain adversarial training procedure than the one proposed in (Ganin et al., 2016).

2 Data preparation

2.1 Dataset augmentation

Training dataset for Subtask B was built using only sentences from source domain. In order to train DANN we take advantage of an additional set of unlabeled sentences from the same domain as a test dataset. We use a subset of data from another corpora (Wachsmuth et al., 2014) consisting of hotel's reviews.

The selection of the subset is as follows, first we take benefit of a "weak" classifier in the form of the baseline, rules-based system provided by the organizers, to predict a class in the mentioned corpora. After that we choose the subset with the same distribution of classes (2085 *suggestions* and 6415 *no suggestions*) and remove too short statements to obtain a histogram of sentences' lengths close to the rest of datasets.

2.2 Preprocessing

We use Keras (Chollet et al., 2015) to perform preprocessing such as removing punctuation signs and lower-casing sentences. Considering that our

Proceedings of the 13th International Workshop on Semantic Evaluation (SemEval-2019), pages 1262–1266
Minneapolis, Minnesota, USA, June 6–7, 2019. ©2019 Association for Computational Linguistics

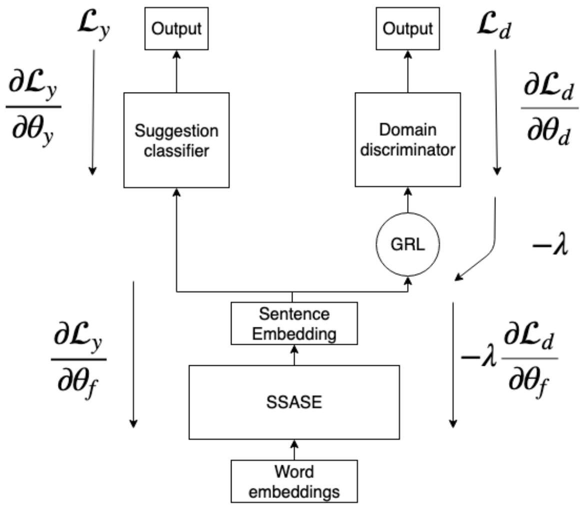

$\mathcal{L}_y$
Output
Output
$\mathcal{L}_d$
$\frac{\partial \mathcal{L}_y}{\partial \theta_y}$
Suggestion classifier
Domain discriminator
$\frac{\partial \mathcal{L}_d}{\partial \theta_d}$
GRL
$-\lambda$
Sentence Embedding
$\frac{\partial \mathcal{L}_y}{\partial \theta_f}$
SSASE
$-\lambda \frac{\partial \mathcal{L}_d}{\partial \theta_f}$
Word embeddings

Figure 1: DANN for cross-domain suggestion mining task.

model is based on recurrent neural network, we also pad sentences or shorten them to have maximum count of 50 tokens. Finally, as the source domain has a lot of urls, we replace all of them with a single *"https"* token.

2.3 Word embeddings

Last step is mapping sentences to mathematically computable form. We leverage the existing language models, which are pre-trained on huge volumes of raw text. Following the recent research, showing superiority of the contextual word embeddings over theirs predecessors, we apply Embeddings from Language Models (ELMo) (Peters et al., 2018) provided by Tensorflow Hub (Abadi et al., 2015). However, we do not fine-tune them with our model.

3 Model description

3.1 Domain-Adversarial Neural Networks

Ganin et al. (2015; 2016) proposed a system for unsupervised Domain Adaptation problem, adaptable to any neural network architecture. It consists of three components: a feature extractor, a label predictor and a domain discriminator. Last one, thanks to gradient reversal layer (GRL), which reverses flow of a gradient with respect to hyperparameter λ, allows to force the feature extractor to learn domain-invariant representations. Fig. 1 presents high level overview of the proposed architecture.

DANN minimizes loss presented in Eq. 1, where y stands for suggestion classifier, d domain descriptor and f for feature extractor presented in the following Section. An upper index in loss $\mathcal{L}$ symbolizes examples' domain.

$$
\begin{aligned}
E(\theta_f,\theta_y,\theta_d) &= \frac{1}{n}\sum_{s=1}^{n}\mathcal{L}_y^s(\theta_f,\theta_y) \\
&- \lambda(\frac{1}{n}\sum_{s=1}^{n}\mathcal{L}_d^s(\theta_f,\theta_d) \\
&+ \frac{1}{n'}\sum_{t=n+1}^{N}\mathcal{L}_d^t(\theta_f,\theta_d))
\end{aligned}
\tag{1}
$$

3.2 Structured Self-Attentive Sentence Embedding

We model sentences using Structured Self-Attentive Sentence Embedding (SSASE) as feature extractor. Taking ELMo word embeddings as an input, followed by Bidirectional LSTM (BiLSTM) layer, SSASE used extended self-attention represented as an attention matrix (A) regularized by a penalization term as in Eq. 2, where $|| \bullet ||_F$ is Frobenius norm of a matrix and I is an identity matrix. Impact of penalization is controlled by hyperparameter α.

$$P = \alpha||(AA^T - I)||_F^2 \tag{2}$$

The attention matrix is calculated as shown in Eq. 3, where H is BiLSTM concatenated output, W_1 and W_2 are matrices of weights.

$$A = softmax(W_2 tanh(W_1 H^T)) \tag{3}$$

3.3 Target Preserving component

To prevent erasing targets domain specific features, we extend DANN with a target domain words decoder (model further referred as TPDANN-SSASE). The decoder is formed by an LSTM layer followed by one fully-connected layer. It takes as input a matching timestep of SSASE's BiLSTM outputs and predicts the input word.

In terms of the objective function, decoder's loss was limited by hyperparameter $\gamma = 0.4$ as in Eq. 4, where θ_r stands for decoder's parameters and θ_f^* - parameters of feature extractor's BiLSTM.

$$
\begin{aligned}
E(\theta_f,\theta_y,\theta_d,\theta_r) &= E(\theta_f,\theta_y,\theta_d) \\
&+ \gamma\frac{1}{n'}\sum_{t=n+1}^{N}\mathcal{L}_r^t(\theta_f^*,\theta_r)
\end{aligned}
\tag{4}
$$

3.4 Training algorithm

We apply a modification of domain-adversarial training procedure (Ganin et al., 2016). We treat an architecture as two separate networks with shared parameters of a feature extractor and a domain descriptor (θ_d and θ_f). In each training step, taking regular DANN as an example, we first update parameters trained using source domain Eq. 5, 6, 7 and then with a target domain examples as shown in Eq. 8, 9, where θ' stands for temporal state of parameters between those two updates and η denotes learning rate. The proposed change in the learning algorithm has been beneficial in terms of exploration properties.

$$\theta'_f \longleftarrow \theta_f - \eta\left(\frac{\partial \mathcal{L}^s_y}{\partial \theta_f} - \lambda \frac{\partial \mathcal{L}^s_d}{\partial \theta_f}\right) \tag{5}$$

$$\theta_y \longleftarrow \theta_y - \eta \frac{\partial \mathcal{L}^s_y}{\partial \theta_y} \tag{6}$$

$$\theta'_d \longleftarrow \theta_d - \eta \frac{\partial \mathcal{L}^s_d}{\partial \theta_d} \tag{7}$$

$$\theta_d \longleftarrow \theta'_d - \eta \frac{\partial \mathcal{L}^t_d}{\partial \theta'_d} \tag{8}$$

$$\theta_f \longleftarrow \theta'_f - \eta\left(-\lambda \frac{\partial \mathcal{L}^t_d}{\partial \theta'_f}\right) \tag{9}$$

4 Evaluation

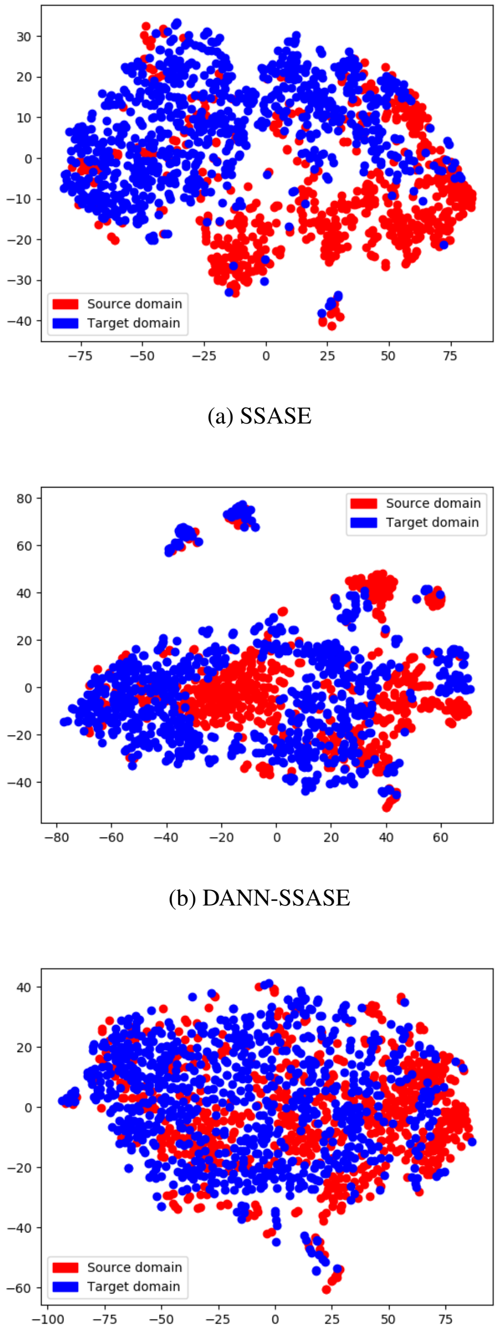
4.1 Results

The metric which was taken into account in SemEval-2019 Task 9 Subtask B was F_1 score. Table 1 presents results for tested architectures for validation and test datasets. Our baseline method is *fastText* (Joulin et al., 2017). It achieves higher score ($F_1 = 0.684$) on Subtask's A validation dataset (source domain) than on target domain, indicating that there is a shift between domains. We notice the same behaviour while testing SSASE model with only a label classifier. By adding domain adaptation components we manage to limit that problem. DANN-SSASE* is trained using default domain-adversarial training procedure (Ganin et al., 2016), while further models benefit from our proposed algorithm. We achieve final score, resulting in the 7^{th} place, by creating unweighted ensemble of three TPDANN-SSASE models.

4.2 Hyperparameters

We use default ELMo embeddings with length of 1024. Each LSTM layer has 300 units (BiLSTM 600). Attention matrix dimensions are accordingly equal to 400 and 9 for W_1 and W_2. We set a penalization hyperparameter α to 0.45.

4.3 Domains shift



(a) SSASE

(b) DANN-SSASE

(c) TPDANN-SSASE

Figure 2: Domains shift reduction between models.

Method	**Validation dataset**	**Test dataset**
fastText	0.532	0.591
SSASE	0.517	0.467
DANN-SSASE*	0.616	0.558
DANN-SSASE	0.781	0.753
TPDANN-SSASE	0.831	0.764
TPDANN-SSASE ensemble	0.836	0.778

Table 1: F_1 score on target domain validation and test datasets.

Method	**Source**	**Target**
SSASE	8.26	8.78
DANN-SSASE	8.08	8.55
TPDANN-SSASE	7.24	7.56

Table 2: Mean count of the 10 nearest neighbours from the same domain. Desired score is equal to 5. To build kNN model, a representation of sentences was extracted from the last layer of a feature extractor and distance was measured using a Euclidean distance.

To measure a problem of domains shift and impact of domain adaptation components in our models we propose a metric based on number of nearest neighbours from the same domain. Assuming that there is no shift between domains, mean number of k nearest neighbours from particular domain over the whole dataset is equal to $\frac{k}{2}$. On the other hand to perfect overlap, it would be equal to k, as each sample could only have neighbours from the same domain.

We take the last layer of a feature extractor as the representations for which euclidean distance metric was employed to find nearest neighbours. Results presented in Tab. 2 indicate that models with better domain-invariant properties have better results in terms of suggestion mining task, TPDANN-SSASE achieves the closest values to $\frac{k}{2}$. In order to present how the domains overlap changed over models, we visualize them using T-SNE (van der Maaten and Hinton, 2008) (Fig. 2). The visualization confirms results presented in Tab. 2 - we observe the highest overlap for TPDANN-SSASE.

5 Conclusion

In this work, we introduced a new system for cross-domain suggestion mining based on the domain-adversarial neural networks. Domains shift reduction led to improvement of classification accuracy in target domain. Our proposed modification of adversarial training procedure allowed ensemble of TPDANN-SASSE models to reach F_1 value of 0.778.

References

Martín Abadi et al. 2015. TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.

François Chollet et al. 2015. Keras. https://keras.io.

Yaroslav Ganin and Victor Lempitsky. 2015. Unsupervised domain adaptation by backpropagation. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 1180–1189, Lille, France. PMLR.

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. 2016. Domain-adversarial training of neural networks. *Journal of Machine Learning Research*, 17(59):1–35.

Tao Gui, Qi Zhang, Haoran Huang, Minlong Peng, and Xuanjing Huang. 2017. Part-of-speech tagging for twitter with adversarial neural networks. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2411–2420. Association for Computational Linguistics.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2017. Bag of tricks for efficient text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association* *for Computational Linguistics: Volume 2, Short Papers*, pages 427–431. Association for Computational Linguistics.

Zhouhan Lin, Minwei Feng, Cicero Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen Zhou, and Yoshua Bengio. 2017. A structured self-attentive sentence embedding.

Bing Liu. 2009. *Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data*. Springer-Verlag, Berlin, Heidelberg.

Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9:2579–2605.

Alicia Martínez Flor. 2005. A theoretical review of the speech act of suggesting: towards a taxonomy for its use in flt. *Revista Alicantina de Estudios Ingleses*.

Sapna Negi and Paul Buitelaar. 2015. Towards the extraction of customer-to-customer suggestions from reviews. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2159–2167, Lisbon, Portugal. Association for Computational Linguistics.

Sapna Negi, Tobias Daudert, and Paul Buitelaar. 2019. Semeval-2019 task 9: Suggestion mining from online reviews and forums. In *Proceedings of the 13th International Workshop on Semantic Evaluation (SemEval-2019)*.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237. Association for Computational Linguistics.

Henning Wachsmuth, Martin Trenkmann, Benno Stein, Gregor Engels, and Tsvetomira Palakarska. 2014. A review corpus for argumentation analysis. In *Computational Linguistics and Intelligent Text Processing*, pages 115–127, Berlin, Heidelberg. Springer Berlin Heidelberg.