INRIA at SemEval-2019 Task 9: Suggestion Mining Using SVM with Handcrafted Features

Ilia Markov
INRIA
Paris, France
ilia.markov@inria.fr

Eric Villemonte De la Clergerie
INRIA
Paris, France
eric.de_la_clergerie@inria.fr

Abstract

We present the INRIA approach to the suggestion mining task at SemEval 2019. The task consists of two subtasks: suggestion mining under single-domain (Subtask A) and cross-domain (Subtask B) settings. We used the Support Vector Machines algorithm trained on handcrafted features, function words, sentiment features, digits, and verbs for Subtask A, and handcrafted features for Subtask B. Our best run archived a F1-score of 51.18% on Subtask A, and ranked in the top ten of the submissions for Subtask B with 73.30% F1-score.

1 Introduction

Suggestion mining can be viewed as a task of extracting suggestions from unstructured text samples (Ramanand et al., 2010; Negi and Buitelaar, 2015). The task goes beyond the sentiment polarity detection and is useful for a variety of purposes, e.g., organizations can improve their products basing on the suggestions from online sources without the need of manually analyzing large amounts of unstructured data (Dong et al., 2013).

In this first edition of the suggestion mining SemEval task (Negi et al., 2019), two settings of the task are addressed: single-domain (or domain-specific) suggestion mining, where the training, development, and test sets belong to the same domain (in the context of this shared task, suggestion forum for Windows platform developers), and cross-domain setting, where training and development/test sets belong to different domains (training on developer suggestion forums and testing on hotel reviews). In the both domains, only explicit expressions of suggestions are considered: lexical cues of a suggestion are explicitly mentioned in the text (Negi et al., 2018).

We approach the task from a machine-learning perspective as a binary classification of given sentences into suggestion and non-suggestion classes. We propose a straightforward approach that can be applied when the availability of training/evaluation data and external linguistic resources is scarce, and evaluate it in the context of this shared task. We were particularly interested in evaluating our approach under cross-domain conditions (Subtask B), since this setting is more common in a real-word scenario of the task.

Further, we briefly describe the datasets used in the competition and focus on the configuration of our system.

2 Data

The training dataset provided by the organizers, as well as the development and test sets for Subtask A consist of explicit suggestion and non-suggestion sentences extracted from the feedback posts on the Universal Windows Platform[1], while the development and test sets for Subtask B are on a different domain: a subset of the sentiment analysis dataset of hotel reviews from the TripAdvisor website (Wachsmuth et al., 2014).

The training, development (dev.), and test datasets statistics in terms of the total number (no.) of sentences, the number of suggestion sentences, and the percentage (%) of suggestion sentences is provided in Table 1. A more detailed description of the datasets used in the shared task can be found in (Negi et al., 2019).

Dataset	Total no. of sentences	No. of suggestions	% of suggestions
Training	8,500	2,085	24.53%
Dev. (Subtask A)	592	296	50.00%
Dev. (Subtask B)	808	404	50.00%
Test (Subtask A)	833	87	10.44%
Test (Subtask B)	824	348	42.23%

Table 1: Suggestion mining datasets statistics.

[1] https://www.uservoice.com

Proceedings of the 13th International Workshop on Semantic Evaluation (SemEval-2019), pages 1204–1207
Minneapolis, Minnesota, USA, June 6–7, 2019. ©2019 Association for Computational Linguistics

As one can see from Table 1, the distribution of the suggestion and non-suggestion classes is balanced in the development sets, but imbalanced in the training and test data, which is closer to the usual distribution of the suggestion sentences in online reviews and forums (Asher et al., 2009; Negi and Buitelaar, 2015; Negi et al., 2018).

3 Methodology

In this section, we describe the features we used and the experimental setup of our best run.

3.1 Features

Handcrafted features Following previous studies on suggestion mining (Ramanand et al., 2010; Brun and Hagège, 2013; Negi and Buitelaar, 2015), we manually selected a list of representative keywords and patterns of a suggestion from the training and development data. It has been shown that suggestion expressions often contain modal verbs (Ramanand et al., 2010), e.g., *should*, *would*, which are included in our list. We also consider some verbs in their infinitive form, e.g., *suggest*, *recommend*, as well as other lexical cues such as comparative adjectives, e.g., *better*, *worse*. For Subtask A, we used a set of 57 handcrafted keywords and 77 keywords were used for Subtask B. Some of the keywords used for Subtask B did not contribute to the results obtained on the subtask A development data, and therefore were discarded. The number of such heuristic keywords in each sentence was used as a feature for the machine-learning algorithm.

Function words Function words are considered one of the most important stylometric features (Kestemont, 2014). We hypothesize that the distribution of function words is different for suggestion and non-suggestion sentences. The function word feature set consists of 318 English function words from the scikit-learn package (Pedregosa et al., 2011). Each function word was considered as a separate feature for Subtask A.

Sentiment features As mentioned in (Brun and Hagège, 2013; Negi et al., 2018), suggestions are usually expressed when a person is not entirely satisfied with the product. To capture this, we used the sentiment information from the NRC Word-Emotion Association Lexicon (Mohammad and Turney, 2013) focusing on words with negative polarity. The number of negative sentiment words in each sentence was used as a feature for Subtask A.

Digits We used the number of digits in a sentence as a feature for Subtask A. This feature is used to evaluate wether the language used in suggestion expressions is more "concrete" (as opposed to abstract) and digits usage can be one of such indicators. Other types of named and numeric entities we examined did not improve our results.

Verbs Following the work by Negi and Buitelaar (2015), we used the number of verbs in a sentence as a feature for Subtask A. The parts-of-speech (POS) tags were obtained using the TreeTagger software package (Schmid, 1995).

When used for Subtask B, function words, sentiment features, digits, and verbs did not improve the performance of our system.

3.2 Experimental setup

Classifier We used the scikit-learn (Pedregosa et al., 2011) implementation of the Support Vector Machines (SVM) algorithm, which is considered among the best-performing algorithms for text classification tasks in general, including when cross-domain conditions and binary classification are concerned (Markov et al., 2017), and for the suggestion mining task in particular (Negi and Buitelaar, 2015; Negi et al., 2016). We set the class_weight parameter to 'balanced' and the penalty parameter (C) to 0.01 for Subtask A and to 0.0001 for Subtask B, tuning the parameters according to the results on the development data.

Weighting scheme We used term frequency (tf) weighting scheme, i.e., the number of times a term occurs in a sentence.

Evaluation For the evaluation of our system, we conducted experiments on the development sets for Subtasks A and B measuring the results in terms of precision, recall, and F1-score for the positive class (the official metric). For training our system, we used only the data provided by the organizers: when evaluating on the development data, we trained our system on the training datasets, while when evaluating on the test data, we merged the training and Subtask A development sets.[2]

[2] Participants were prohibited from using additional hand-labeled training data of the same domain for Subtask B.

4 Results and discussion

First, we present the results in terms of precision (%), recall (%), and on F1-score for the positive class (%) obtained on the Subtask A development data. The contribution of each feature type incorporated in our system is shown through an ablation study in Table 2. The number of features (No.) is also provided.[3] The handcrafted features and function words are the most indicative features in our system (when used in isolation they achieve a F1-score of 72.76% and 69.77%, respectively), while other types of features slightly improve the performance of our system.

Features	Precision	Recall	F1-score	No.
All features	77.93	78.72	78.32	275
– handcrafted	75.00	66.89	70.71	274
Drop:	**2.93**	**11.83**	**7.61**	
– function words	71.72	71.96	71.84	4
Drop:	**6.21**	**6.76**	**6.48**	
– sentiment features	77.29	77.03	77.16	274
Drop:	**0.64**	**1.69**	**1.16**	
– digits	77.52	78.04	77.78	274
Drop:	**0.41**	**0.68**	**0.54**	
– verbs	77.67	78.72	78.19	274
Drop:	**0.26**	**0.00**	**0.13**	

Table 2: Ablation study of the feature types used for Subtask A.

The results in terms of precision, recall, and F1-score on the development sets for Subtasks A and B, as well as the official results obtained on the test sets are provided in Table 3. The results for the rule-based baseline approach proposed by the organizers are also presented.

Subtask A	Precision	Recall	F1-score
Baseline dev.	58.72	93.24	72.06
Our dev.	77.93	78.72	78.32
Baseline test	15.66	91.95	26.76
Our test	38.92	74.71	51.18
Subtask B	**Precision**	**Recall**	**F1-score**
Baseline dev.	72.85	81.68	77.01
Our dev.	85.42	82.67	84.03
Baseline test	68.86	78.16	73.22
Our test	73.62	72.99	73.30

Table 3: Results for the INRIA and the baseline approaches on the development (dev.) and test sets for Subtasks A and B.

Though the F1-score achieved by our system is higher than the one achieved by the official baselines in all cases, there is a considerable drop on the test sets: 27.14% F1-score drop for Subtask A and 10.73% for Subtask B. For Subtask A, the drop is mainly caused by the low precision achieved on the test set (precision of 77.93% on the development set and 38.92% on the test set).

After the evaluation period, in order to examine whether the drop in precision and the large number of false positives provided by our system on the Subtask A test set is partly related to the different distribution of classes in the development and test data – 50% and 10.44% of suggestions, respectively (see Table 1) –, we balanced the classes in the training and test sets to be in phase with each other and evaluated the impact of classes distribution on the results achieved by our system:

- *Test-like distribution*: we randomly removed positive examples from the training data so that the distribution of classes in the training set is the same as in the test set (10.44% of positive examples instead of 26.19% in the merged training and Subtask A development data).

- *Train-like distribution*: we removed negative examples from the test data so that the distribution of positive classes in the test set is the same as in the training data (26.19% instead of 10.44%).

The results for these two experiments are shown in Tables 4 and 5.[4]

Setting	Precision	Recall	F1-score
Original distribution	38.92	74.71	51.18
Test-like distribution	41.96	75.86	54.03
Gain:	**3.04**	**1.15**	**2.85**

Table 4: Results for the original and test-like distributions of positive classes.

Setting	Precision	Recall	F1-score
Original distribution	38.92	74.71	51.18
Train-like distribution	64.87	74.71	69.07
Gain:	**25.95**	**0.00**	**17.89**

Table 5: Results for the original and train-like distributions of positive classes.

[3] Note that we use function words as features (274 features), while the number of occurrences of the handcrafted keywords, sentiment features, digits, and verbs is considered as a feature (4 features in total).

[4] The result for the test-/train-like distributions was calculated as average over three experiments removing three different sets of positive/negative examples.

As one can see from Tables 4 and 5, balancing the distribution of positive classes, so that it is the same in the training and the evaluation data, enhances the performance of our system (by around 3% in the test-like setting and around 18% in the train-like setting) mainly due to the increase in precision, which indicates that the distribution of calsses should be taken into account when developing a robust suggestion mining system.

5 Conclusions

We presented the description of the best submission of the INRIA team to the suggestion mining shared task at SemEval 2019. Our approach is based on the Support Vector Machines algorithm trained on handcrafted features, function words, sentiment features, digits, and verbs for Subtask A (single-domain setting). For Subtask B (cross-domain setting), only handcrafted features are used. Our best run showed 51.18% F1-score for Subtask A and 73.30% for Subtask B. The results obtained on the test sets are lower than on the development data. Additional experiments revealed that the drop in F1-score is partly related to the different distribution of classes in the training data and in the development set used to evaluate and tune our system. In future work, we plan to improve our list of handcrafted features to make our system robust to variations in the distribution of classes and across different suggestion mining domains.

References

Nicholas Asher, Farah Benamara, and Yannick Mathieu. 2009. Appraisal of opinion expressions in discourse. *Lingvistic Investigationes*, 31:279–292.

Caroline Brun and Caroline Hagège. 2013. Suggestion mining: Detecting suggestions for improvement in users' comments. *Research in Computing Science*, 70:199–209.

Li Dong, Furu Wei, Yajuan Duan, Xiaohua Liu, Ming Zhou, and Ke Xu. 2013. The automated acquisition of suggestions from tweets. In *Proceedings of the 27th AAAI Conference on Artificial Intelligence*, Bellevue, Washington, USA. AAAI Press.

Mike Kestemont. 2014. Function words in authorship attribution. From black magic to theory? In *Proceedings of the 3rd Workshop on Computational Linguistics for Literature*, pages 59–66, Gothenburg, Sweden. ACL.

Ilia Markov, Helena Gómez-Adorno, Grigori Sidorov, and Alexander Gelbukh. 2017. The winning approach to cross-genre gender identification in Russian at RUSProfiling 2017. In *FIRE 2017 Working Notes*, volume 2036, pages 20–24, Bangalore, India. CEUR-WS.org.

Saif Mohammad and Peter Turney. 2013. Crowdsourcing a word-emotion association lexicon. *Computational Intelligence*, 29:436–465.

Sapna Negi, Kartik Asooja, Shubham Mehrotra, and Paul Buitelaar. 2016. A study of suggestions in opinionated texts and their automatic detection. In *Proceedings of the Fifth Joint Conference on Lexical and Computational Semantics*, pages 170–178, Berlin, Germany. ACL.

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, Association for Computational Linguistics*, pages 2159–2167, Lisbon, Portugal. ACL.

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)*.

Sapna Negi, Maarten de Rijke, and Paul Buitelaar. 2018. Open domain suggestion mining: Problem definition and datasets. *arXiv preprint arXiv:1806.02179*.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Jaiprakash Ramanand, Krishna Bhavsar, and Niranjan Pedanekar. 2010. Wishful thinking – finding suggestions and 'buy' wishes from product reviews. In *Proceedings of the NAACL HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text*, pages 54–61, Los Angeles, California, USA. ACL.

Helmut Schmid. 1995. Improvements in part-of-speech tagging with an application to German. In *Proceedings of the ACL SIGDAT-Workshop*, pages 47–50, Dublin, Ireland. ACL.

Henning Wachsmuth, Martin Trenkmann, Benno Stein, Gregor Engels, and Tsvetomira Palakarska. 2014. A review corpus for argumentation analysis. In *Proceedings of the 15th International Conference on Computational Linguistics and Intelligent Text Processing*, volume 8404, pages 115–127, Kathmandu, Nepal. Springer.