A Study of Suggestions in Opinionated Texts and their Automatic Detection

Sapna Negi[1] **Kartik Asooja**[1] **Shubham Mehrotra**[1,2] **Paul Buitelaar**[1]

[1] Insight Centre for Data Analytics, National University of Ireland, Galway
firstname.lastname@insight-centre.org

[2] Indian Institute of Information Technology Allahabad, India
shubhammehrotra94@gmail.com

Abstract

We study the automatic detection of suggestion expressing text among the opinionated text. The examples of such suggestions in online reviews would be, customer suggestions about improvement in a commercial entity, and advice to the fellow customers. We present a qualitative and quantitative analysis of suggestions present in the text samples obtained from social media platforms. Suggestion mining from social media is an emerging research area, and thus problem definition and datasets are still evolving; this work also contributes towards the same. The problem has been formulated as a sentence classification task, and we compare the results of some popular supervised learning approaches in this direction. We also evaluate different kinds of features with these classifiers. The experiments indicate that deep learning based approaches tend to be promising for this task.

1 Introduction

Online text is becoming an increasingly popular source for acquiring public opinions towards entities like persons, products, services, brands, events, etc. The area of opinion mining focuses on exploiting this abundance of opinions, by mainly performing sentiment based summarisation of text into *positive, negative, and neutral* categories, using sentiment analysis methods. In addition to the online reviews and blogs, people are increasingly resorting to social networks like Twitter, Facebook etc. to instantly express their sentiments and opinions about the products and services they might be experiencing at a given time.

On a closer look, it is noticeable that opinionated text also contains information other than sentiments. This can be validated from the presence of large portions of *neutral* or *objective* or *non-relevant* labelled text in state of the art sentiment analysis datasets. One such information type is suggestions. Table 1 shows the instances of suggestions in sentiment analysis datasets which were built on online reviews. These suggestions may or may not carry positive or negative sentiments towards the reviewed entity. In the recent past, suggestions have gained the attention of the research community, mainly for industrial research, which led to the studies focussing on suggestion detection in reviews (Ramanand et al., 2010; Brun and Hagege, 2013).

The setting up of dedicated suggestion collection forums by brand owners, shows the importance of suggestions for the stakeholders. Therefore, it would be useful if suggestions can be automatically extracted from the large amount of already available opinions. In the cases of certain entities where suggestion collection platforms [1] are already available and active, suggestion mining can be used for summarisation of posts. Often, people tend to provide the context in such posts, which gets repetitive in the case of large number of posts, suggestion mining methods can extract the exact sentence in the post where a suggestion is expressed.

This task has so far been presented as a binary classification of sentences, where the available opinionated text about a certain entity is split into sentences and these sentences are then classified as suggestions or non-suggestions. The previous studies were carried out in a limited scope, mainly for specific domains like reviews, focusing on one use case at a time. The path to the leaf

[1] https://feedly.uservoice.com/forums/192636-suggestions/category/64071-mobile

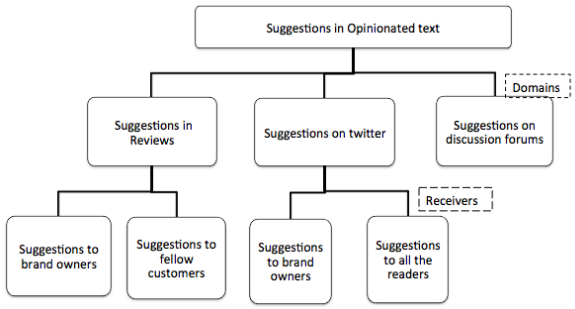

Figure 1: Problem scopes in suggestion detection

nodes in Figure 1 summarises the scope of suggestion mining studies so far. These studies developed datasets for individual tasks and domains, and trained and evaluated classifier models on the same datasets.

We analyse manually labelled datasets from different domains, including the existing datasets, and the datasets prepared by us. The ratio of suggestion and non-suggestion sentences vary across domains, where the datasets from some domains are too sparse for training statistical classifiers. We also introduce two datasets which are relatively richer in suggestions. In Table 1 we report similar linguistic nature of suggestions across these datasets, which presses for domain independent approaches. Therefore, as a deviation from previous studies, this work investigates the generalisation of the problem of suggestion detection i.e. the detection of all suggestions under the root node in Figure 1.

In this work, we compare different methods of suggestion mining using all available datasets. These include manually crafted rules, Support Vector Machines (SVM) with proposed linguistic features, Long Short Term Memory (LSTM) Neural Networks, and Convolutional Neural Networks (CNN). We also compare the results from these approaches with the previous works whose datasets are available. We also perform cross-domain train test experiments. With most of the datasets, Neural Networks (NNs) outperform SVM with the proposed features. However, the overall results for out of domain training remain low. We also compare two different types of word embeddings to be used with the NNs for this task.

2 Problem Definition and Scope

As stated previously, the task of suggestion detection has been framed as binary classification of sentences into *suggestion* (positive class) and *non-suggestion* (negative class).
We previously provided a fine grained problem definition (Negi and Buitelaar, 2015) in order to prepare benchmark datasets and ensure consistency in future task definitions. We identified three parameters which define a suggestion in the context of opinion mining: receiver of suggestion, textual unit of suggestion, and the type of suggestion in terms of its explicit or implicit nature.

While the unit of suggestion still remains as sentence in this work, and the type as explicit expression, we aim for the evaluation of different classifier models for the detection of any suggestion from any opinionated text. The motivation lies in our observation that explicitly expressed suggestions appear in similar linguistic forms irrespective of domain, target entity, and the intended receiver (Table 1). Furthermore, datasets used by the previous studies indicate that aiming the detection of specific suggestions restricts the annotations to suggestions of a specific type, which in turn aggravates class imbalance problem in the datasets (Table 2). It also renders these datasets unsuitable for a generic suggestion detection task, since the negative instances may also comprise of suggestions, but not of the desired type.

3 Related Work

In the recent years, experiments have been performed to automatically detect sentences which contain suggestions. Targeted suggestions were mainly the ones which suggest improvements in a commercial entity. Therefore, online reviews remains the main focus, however, there are a limited number of works focussing on other domains too.

Suggestions for product improvement: Studies like Ramanand et al. (2010) and Brun et al. (2013) employed manually crafted linguistic rules to identify suggestions for product improvement. The evaluation was performed on a small dataset (∼60 reviews). Dong et al. (2013) performed classification of given tweets about Microsoft Windows' phone as suggestions for improvement or not. They compared SVM and Factorisation Machines (FM) based classifiers. For features,

Source, Entity/Topic	Sentence	Sentiment Label	Intended Receiver	Linguistic Properties
Reviews, Electronics	I would recommend doing the upgrade to be sure you have the best chance at trouble free operation.	Neutral	Customer	Subjunctive, Imperative, lexical clue: *recommend*
Reviews, Electronics	My one recommendation to creative is to get some marketing people to work on the names of these things	Neutral	Brand owner	Imperative, lexical clue: *recommendation*
Reviews, Hotels	Be sure to specify a room at the back of the hotel.	Neutral	Customer	Imperative
Reviews, Hotel	The point is, don't advertise the service if there are caveats that go with it.	Negative	Brand Owner	Imperative
Tweets, Windows Phone	Dear Microsoft, release a new zune with your wp7 launch on the 11th. It would be smart	Neutral	Brand owner	Imperative, subjunctive
Discussion thread, Travel	If you do book your own airfare, be sure you don't have problems if Insight has to cancel the tour or reschedule it	Neutral	Thread participants	Conditional, imperative
Tweets, open topics	Again I'm reminded of some of the best advice I've ever received: thank you notes. Always start with the thank you notes.	NA	General public	Imperative, Lexical clue: *a*dvice
Suggestion forum, Software	Please provide consistency throughout the entire Microsoft development ecosystem!	NA	Brand owner	Imperative, lexical clue: *please*

Table 1: Examples of suggestions from different domains, about different entities and topics, and intended for different receivers. Sentiment labels are the sentiment towards a reviewed entity, if any.

they used certain hash tags and mined frequently appearing word based patterns from a separate dataset of suggestions about Microsoft phones.

Suggestions for fellow customers: In one of our previous works(Negi and Buitelaar, 2015), we focussed on the detection of those suggestions in reviews which are meant for the fellow customers. An example of such suggestion in a hotel review is, *If you do end up here, be sure to specify a room at the back of the hotel*. We used SVM classifier with a set of linguistically motivated features. We also stressed upon the highly subjective nature of suggestion labelling task, and thus performed a study of a formal definition of suggestions in the context of suggestion mining. We also formulated annotation guidelines, and prepared a dataset for the same.

Advice Mining from discussion threads: Wicaksono et al. (2013) detected advice containing sentences from travel related discussion threads. They compared sequential classifiers based on Hidden Markov Model (HMM) and Conditional Random Fields (CRF), considering each thread as a sequence of sentences labelled as advice and non-advice. They also some features which were dependent on the position of a sentence in its thread. This approach was therefore specific to the domain of discussion threads. Their annotations seem to consider implicit expressions of advice as *advice*.

Text Classification using deep learning: Recently NNs are being effectively used for text classification tasks, like sentiment classification and semantic categorisation. LSTM (Graves, 2012), and CNN (Kim, 2014a) are the two most popular neural network architectures in this regard.

Tweet classification using deep learning: To the best of our knowledge, deep learning has only been employed for sentiment based classification of tweets. CNN (Severyn and Moschitti, 2015) and LSTM (Wang et al., 2015) have demonstrated good performance in this regard.

4 Datasets

The required datasets for this task are a set of sentences obtained from opinionated texts, which are labelled as *suggestion* and *non-suggestion*, where suggestions are explicitly expressed.

Existing Datasets: Datasets from most of the previous studies on suggestions for product improvement are unavailable due to their industrial ownership. The currently available datasets are:

1) Twitter dataset about Windows phone: This dataset comprises of tweets which are addressed to Microsoft. The tweets which expressed suggestions for product improvement are labelled as suggestions (Dong et al., 2013). Due to the

short nature of tweets, suggestion detection is performed on the tweet level, rather than the sentence level. The authors indicated that they have labeled the explicit expressions of suggestions in the dataset.
2) Electronics and hotel reviews: A review dataset, where only those sentences which convey suggestions to the fellow customers are considered as suggestions (Negi and Buitelaar, 2015).
3) Travel advice dataset: Obtained from travel related discussion forums. All the advice containing sentences are tagged as *advice* (Wicaksono and Myaeng, 2013). One problem with this dataset is that the statements of facts (*implicit suggestions*) are also tagged as advice, for example, *The temperature may reach upto 40 degrees in summer*.

Introduced Datasets: In this work, we identify additional sources for suggestion datasets, and prepare labelled datasets with larger number of explicitly expressed suggestions.
1) Suggestion forum: Posts from a customer support platform[2] which also hosts dedicated suggestion forums for products. Though most of the forums for commercial products are closed access, we discovered two forums which are openly accessible: Feedly mobile app[3], and Windows app studio[4]. We collected samples of posts for these two products. Posts were then split into sentences using the sentence splitter from Stanford CoreNLP toolkit (Manning et al., 2014). Two annotators were asked to label 1000 sentences, on which the inter-annotator agreement (kappa) of 0.81 was obtained. Rest of the dataset was annotated by only one annotator. Due to the annotation costs, we limited the size of data sample, however this dataset is easily extendible due to the availability of much larger number of posts on these forums.
2) We also prepared a tweet dataset where tweets are a mixture of random topics, and not specific to any given entity or topic. These tweets were collected using the hashtags *suggestion, advice, recommendation, warning*, which increased the chance of appearance of suggestions in this dataset. Due to the noisy nature of tweets, two annotators performed annotation on all the tweets. The inter-annotator agreement was calculated as 0.72. Only those tweets were retained for which the annotators agreed on the label.
3) We also re-tagged the travel advice dataset from Wicaksono et al. (2013) where only those suggestions which were explicitly expressed were retained as suggestions.

Table 2 details all the available datasets including the ones we are introducing in this work. The introduced datasets contain higher percentage of suggestions. We therefore train models on the introduced datasets, and evaluate them on the existing datasets.

Dataset	**Suggestion Type**	**Suggestions/ Total Instances**
Existing Datasets		
Electronics Reviews, (Negi and Buitelaar, 2015)	Only for customers, explicitly expressed	324/3782
Hotel Reviews, (Negi and Buitelaar, 2015)	Only for customers, explicitly expressed	448/7534
Tweets Microsoft phone, (Dong et al., 2013)	Only for brand owners, explicitly expressed	238/3000
Travel advice 1, (Wicaksono and Myaeng, 2013)	Any suggestion, explicitly or implicitly expressed	2192/5199
Introduced Datasets		
Travel advice 2 (Re-labeled Travel advice 1)	Any suggestion, explicitly expressed	1314/5183
Suggestion forum[5]	Any suggestion, explicitly expressed	1428/5724
Tweets with hashtags: *suggestion, advice, recommendation, warning*	Any suggestion, explicitly expressed	1126/4099

Table 2: Available suggestion detection datasets

5 Automatic Detection of Suggestions

Some of the conventional text classification approaches have been previously studied for this task, primarily, rules and SVM classifiers. Each approach was only evaluated on the datasets prepared within the individual works. We employ these two approaches on all the available datasets for all kinds of suggestion detection task. We then perform a study of the employability of LSTM and CNN for this kind of text classification task. We evaluate all the statistical classifiers in both domain dependent and independent training. The results demonstrate that deep learning methods have

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

[3]https://feedly.uservoice.com/forums/192636-suggestions

[4]https://wpdev.uservoice.com/forums/110705-universal-windows-platform

an advantage over the conventional approaches for this task.

5.1 Rule based classification

This approach uses a set of manually formulated rules aggregated from the previous rule based experiments (Ramanand et al., 2010; Goldberg et al., 2009). These rules exclude the rules provided by Brun et al. (2013), because of their dependency on in-house (publicly unavailable) components from Brun et al. (2013). Only those rules have been used which do not depend on any domain specific vocabulary. A given text is labeled as a suggestion, if at least one of the rules is true.

1. Modal verbs (MD) followed by a base form of verb (VB), followed by an adjective.
2. At-least one clause starts with a present tense of verb (VB, VBZ, VBP). This is a naive method for detecting imperative sentences. Clauses are identified using the parse trees; the sub-trees under S and SBAR are considered as clauses.
3. Presence of any of the suggestion keywords/phrases *suggest, recommend, hopefully, go for, request, it would be nice, adding, should come with, should be able, could come with, i need, we need, needs, would like to, would love to*.
4. Presence of templates for suggestions expressed in the form of wishes [*would like *(if), I wish, I hope, I want, hopefully, if only, would be better if, *(should)*, would that, can't believe .*(didn't).*, (don't believe).*(didn't), (do want), I can has*].

The part of speech tagging and parsing is performed using Stanford parser (Manning et al., 2014). Table 3 shows the results of rule based classification for the positive class i.e. suggestion class. With the available datasets, detection of negative instances is always significantly better than the positive ones, due to class imbalance.

Dataset	Prec.	Rec.	F1
Electronics Reviews	0.229	0.660	0.340
Hotel Reviews	0.196	0.517	0.285
Travel discussion 2	0.312	0.378	0.342
Microsoft Tweets	0.207	0.756	0.325
New Tweets	0.200	0.398	0.266
Suggestion Forum	0.461	0.879	0.605

Table 3: Results of Suggestion Detection using rule based classifier. Reported metrics are only for the suggestion class.

5.2 Statistical classifiers

SVM was used in almost all the related work either as a proposed classifier with some feature engineering, or for comparison with other classifiers.

Support Vector Machines: SVM classifiers are popularly used for text classification in the research community. We perform the evaluation of a classifier using SVM with the standard n-gram features (uni, bi-grams) and the features proposed in our previous work (Negi and Buitelaar, 2015). These features are sequential POS patterns for imperative mood, sentence sentiment score obtained using SentiWordNet, and information about *nsubj* dependency present in the sentence. We use LibSVM[6] implementation with the parameters specified previously in Negi and Buitelaar (2015). No oversampling is used, instead class weighting is applied by using class weight ratio depending upon the class distribution of the negative and positive class respectively in the training dataset.

Deep Learning based classifiers: Recent findings about the impressive performance of deep learning based models for some of the natural language processing tasks calls for similar experiments in suggestion mining. We therefore present the first set of deep learning based experiments for the same. We experiment with two kinds of neural network architectures: LSTM and CNN. LSTM effectively captures sequential information in text, while retaining the long term dependencies. In a standard LSTM model for text classification, text can be fed to the input layer as a sequence of words, one word at a time. Figure 2 shows the architecture of LSTM neural networks for binary text classification.
On the other hand, CNN is known to effectively capture local co-relations of spatial or temporal structures, therefore a general intuition is that CNN might capture well the good n-gram features at different positions in a sentence.

5.3 Features

Features for SVM: The feature evaluation of (Negi and Buitelaar, 2015) indicated that POS tags, certain keywords (lexical clues), POS

[6]https://www.csie.ntu.edu.tw/ cjlin/libsvm/

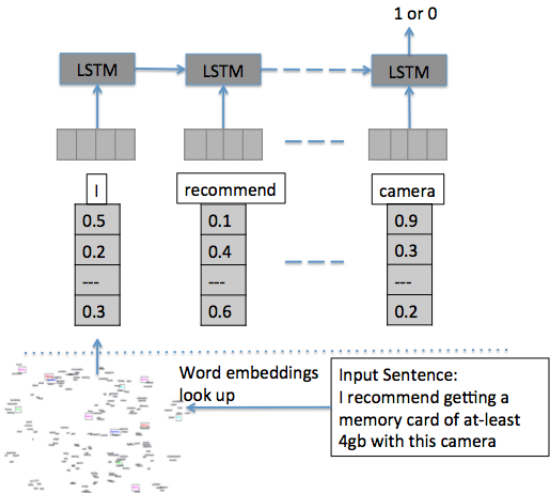

Figure 2: Architecture for using LSTM as a binary text classifier

patterns for imperative mood, and certain dependency information about the subject, can be useful features for the detection of suggestions. In the previous works, the feature types were manually determined. We now aim to eliminate the need of manual determination of feature types. A recently popular approach of doing this is to use neural networks with *word embeddings* (Bengio et al., 2003) based feature vectors, instead of using classic count-based feature vectors.

Word embeddings for Neural Networks: In simpler terms, word embeddings are automatically learnt vector representations for lexical units. Baroni et al. (2014) compared the word embeddings obtained through different methods, by using them for different semantic tasks. Based on those comparisons, we use a pre-trained COMPOSES[7] embeddings, which were developed by Baroni et al. (2014). These embeddings/word vectors are of size 400. For experiments on twitter datasets, we used Glove (Pennington et al., 2014) based word embeddings learnt on Twitter data[8], which comprises of 200 dimensions.
We additionally experiment with dependency based word embeddings (Deps)[9] (Levy and Goldberg, 2014). These embeddings determine the context of a word on the basis of linguistic dependencies, instead of window based context used by COMPOSES. Therefore, Deps tends to perform better in determining the functional similarity between words, as compared to COMPOSES.
Additional feature for NNs: For neural network based classifiers, we also experimented with POS tags as an additional feature with the pre-trained word embeddings. This tends to decrease the precision and increase the recall, but results in an overall decrease of F-1 score in most of the runs. Therefore, we do not report the results of these experiments.

5.4 Configurations

NN Configuration: Considering the class imbalance in the datasets, we employ oversampling of the minority class (positive) to adjust the class distribution of training data. While performing cross validation, we perform oversampling on training data for each fold separately after cross-validating.
LSTM: For LSTM based classification, we use 2 hidden layers of 100 and 50 neurons respectively, and 1 softmax output layer. We also utilize L2 regularization to counter overfitting. For LSTMs, we use the softsign activation function.
CNN: We used a filter window of 2 with 40 feature maps in CNN, thus giving 40 bigram based filters (Kim, 2014b). A subsampling layer with max pooling is used.

In-Domain and Cross-Domain Evaluation: In the case of statistical classifiers, we perform the experiments in two sets. The first set of experiments (Table 4, 6) evaluate a classifier (and feature types) for the cases where labeled data is available for a specific domain, entity, or receiver specific suggestions. In this case, evaluation is performed using a 10 fold cross validation with SVM and 5 fold with NN classifiers. The second set of experiments evaluate the classifiers (and feature types) for a generic suggestion detection task, where the model can be trained on any of the available datasets. These experiments evaluate the classifier algorithms, as well as the training datasets. In the case of twitter, training is performed on twitter dataset, while evaluation for this cross-domain setting is performed on the Microsoft tweet dataset.

[7] Best predict vectors on http://clic.cimec.unitn.it/composes/semantic-vectors.html
[8] http://nlp.stanford.edu/projects/glove/
[9] Dependency-Based on https://levyomer.wordpress.com/2014/04/25/dependency-based-word-embeddings/

Dataset	LSTM		CNN	
	COMP.	Deps	COMP.	Deps
Hotel	**0.638**	0.607	**0.578**	0.550
Electronics	**0.672**	0.608	**0.611**	0.556
Travel advice 2	0.617	**0.625**	**0.586**	0.564
Sugg Forum	**0.752**	0.732	**0.714**	0.695

Table 6: F-1 score for the suggestion class, using *COMPOSES* and *Deps* embeddings with LSTM and CNN. 5 fold cross validation.

Pre-processing: We also compared experiments on tweets with pre-processing, and without pre-processing the tweets. The pre-processing involved removing URLs and hashtags, and normalisation of punctuation repetition. Pre-processing tends to decrease the performance in all the experiments. Therefore, none of the experiments reported by us use pre-processing on tweets.

6 Results and Discussions

Tables 4, 7 show the Precision, Recall and F-1 score for the suggestion class (positive class). In general, rule based classifier shows a higher recall, but very low precision, leading to very low F-1 scores as compared to statistical classifiers, where LSTM emerges as a winner in majority of the runs. Below we summarise different observations from the results.

Embeddings: COMPOSES embeddings prove to be a clear winner in our experiments. Deps outperform COMPOSES in only 3 cases out of all the experiments reported in Tables 6, 8. It was observed that using Deps always resulted in higher recall, however F-1 scores dropped due to a simultaneous drop in precision. Also, Deps embeddings tend to perform better with LSTM, as compared to CNN.

Comparison with Related Work: Table 5 compares the results from those works whose datasets are available. It shows that LSTM outperforms the best results from Wicaksono et al. by a small margin, provided that they used features which are only valid for dicussion threads, while the LSTM uses generic features (embeddings). The table also shows a comparison of other approaches with the factorization machine based approach adopted by Dong et al. (2013) for classifying Microsoft tweets, which provides a much higher F-1 score. This can be attributed to the use of fine tuning (oversampling, thresholding) for the class imbalance problem. Dong et al. also report results using FM and SVM which do not use fine tuning; those results are in line with our SVM and LSTM results. Additionally, they also use hashtags and suggestion templates extracted from an unavailable dataset of suggestions for Microsoft phones.

Train/Test	LSTM		CNN	
	COMP	Deps	COMP	Deps
Sugg-Forum/Hotel	**0.450**	0.38	0.363	**0.367**
Sugg-Forum/Electronics	**0.510**	0.470	**0.393**	0.384
Sugg-Forum/Travel Advice	0.323	**0.340**	**0.453**	0.330
Travel advice/Hotel	0.316	**0.349**	**0.304**	0.292

Table 8: Evaluation of *COMPOSES* and *Dependency* embeddings with LSTM and CNN in a cross domain train-test setting.

SVM versus NNs: In most cases, the neural network based classifiers outperformed SVM, see tables 4, 7. Although SVM in combination with feature engineering and parameter tuning, proves to be a competent alternative, specially with the more balanced new datasets. The newly introduced datasets (suggestions about Feedly app and Windows platform) produce better results than the existing sparse datasets for the in-domain evaluation, see table 4. This can be again attributed to the better class representation in this dataset.

Text type: The results of tweet datasets in general show much lower classification accuracy than the datasets of standard texts for cross domain training, see table 7. In the case of in-domain evaluation for the Microsoft tweet dataset, SVM performs better than neural networks, and vice versa in the case of the new tweet dataset, see table 4.

7 Conclusion and Future Work

In this work, we presented an insight into the problem of suggestion detection, which extracts different kinds of suggestions from opinionated text. We point to new sources of suggestion rich datasets, and provide two additional datasets which contain larger number of suggestions as

compared to the previous datasets. We compare various approaches for suggestion detection, including the ones used in the previous works, as well as the deep learning approaches for sentence classification which have not yet been applied to this problem.

Since suggestions tend to exhibit similar linguistic nature, irrespective of topics and intended receiver of the suggestions, there is a scope of learning domain independent models for this task. Therefore, we apply the discussed approaches both in a domain dependent, and domain independent setting, in order to evaluate the domain independence of the proposed models.

Neural networks in general performed better, in both in-domain and cross-domain evaluation. The initial results for domain independent training are poor. In light of the findings from this work, domain transfer approaches would be an interesting direction for future works in this problem.

The results also point out the challenges and complexity of the task. Preparing datasets where suggestions are labeled at a phrase or clause level might reduce the complexities arising due to long sentences.

Acknowledgement

This work has been funded by the European Unions Horizon 2020 programme under grant agreement No 644632 MixedEmotions, and the Science Foundation Ireland under Grant Number SFI/12/RC/2289 (Insight Center).

References

[Baroni et al.2014] Marco Baroni, Georgiana Dinu, and Germán Kruszewski. 2014. Don't count, predict! a systematic comparison of context-counting vs. context-predicting semantic vectors. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 238–247, Baltimore, Maryland, June. Association for Computational Linguistics.

[Bengio et al.2003] Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. 2003. A neural probabilistic language model. *J. Mach. Learn. Res.*, 3:1137–1155, March.

[Brun and Hagege2013] C. Brun and C. Hagege. 2013. Suggestion mining: Detecting suggestions for improvements in users comments. In *Proceedings of 14th International Conference on Intelligent Text Processing and Computational Linguistics*.

[Dong et al.2013] Li Dong, Furu Wei, Yajuan Duan, Xiaohua Liu, Ming Zhou, and Ke Xu. 2013. The automated acquisition of suggestions from tweets. In Marie desJardins and Michael L. Littman, editors, *AAAI*. AAAI Press.

[Goldberg et al.2009] Andrew B. Goldberg, Nathanael Fillmore, David Andrzejewski, Zhiting Xu, Bryan Gibson, and Xiaojin Zhu. 2009. May all your wishes come true: A study of wishes and how to recognize them. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, NAACL '09, pages 263–271, Stroudsburg, PA, USA. Association for Computational Linguistics.

[Graves2012] Alex Graves. 2012. Supervised sequence labelling. In *Supervised Sequence Labelling with Recurrent Neural Networks*, pages 5–13. Springer Berlin Heidelberg.

[Kim2014a] Yoon Kim. 2014a. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*.

[Kim2014b] Yoon Kim. 2014b. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751, Doha, Qatar, October. Association for Computational Linguistics.

[Levy and Goldberg2014] Omer Levy and Yoav Goldberg. 2014. Dependency-based word embeddings. In *ACL*.

[Manning et al.2014] Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*, pages 55–60.

[Negi and Buitelaar2015] 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, September. Association for Computational Linguistics.

[Pennington et al.2014] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

[Ramanand et al.2010] J 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, CA, June. Association for Computational Linguistics.

[Severyn and Moschitti2015] Aliaksei Severyn and Alessandro Moschitti. 2015. Unitn: Training deep convolutional neural network for twitter sentiment classification. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 464–469, Denver, Colorado, June. Association for Computational Linguistics.

[Wang et al.2015] Xin Wang, Yuanchao Liu, Chengjie SUN, Baoxun Wang, and Xiaolong Wang. 2015. Predicting polarities of tweets by composing word embeddings with long short-term memory. 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 1343–1353, Beijing, China, July. Association for Computational Linguistics.

[Wicaksono and Myaeng2013] Alfan Farizki Wicaksono and Sung-Hyon Myaeng. 2013. Automatic extraction of advice-revealing sentences foradvice mining from online forums. In *K-CAP*, pages 97–104. ACM.

[Zhou et al.2015] Chunting Zhou, Chonglin Sun, Zhiyuan Liu, and Francis C. M. Lau. 2015. A C-LSTM neural network for text classification. *CoRR*, abs/1511.08630.

Data	Precision			Recall			F1 score		
	SVM	LSTM	CNN	SVM	LSTM	CNN	SVM	LSTM	CNN
Hotel	**0.580**	0.576	0.505	0.512	**0.717**	0.703	0.543	**0.639**	0.578
Electronics	0.645	**0.663**	0.561	0.621	**0.681**	0.671	0.640	**0.672**	0.612
Travel advice 2	0.458	**0.609**	0.555	**0.732**	0.630	0.621	0.566	**0.617**	0.586
Microsoft Tweets	0.468	**0.591**	0.309	**0.903**	0.514	0.766	**0.616**	0.550	0.441
New tweets	**0.693**	0.619	0.590	0.580	0.674	**0.752**	0.632	0.645	**0.661**
Suggestion forum	0.661	**0.738**	0.665	0.760	0.716	**0.772**	0.712	**0.727**	0.713

Table 4: In-domain training: Performance of SVM (10 fold), LSTM, and CNN (5 fold) using cross validation on the available datasets. The listed results are for the suggestion class only. SVM uses features from Negi and Buitelaar (2015), and neural networks use pre-trained word embeddings (COMPOSES for normal text and Twitter Glove for tweets).

Dataset	Related work	F1 type	F1 (Related Work)	SVM	LSTM	CNN
Travel Advice 1	(Wicaksono and Myaeng, 2013)	Weighted F-1 score for both classes	0.756	0.680	**0.762**	0.692
Microsoft tweets	(Dong et al., 2013)	F-1 score for suggestions only	**0.694**	0.616	0.550	0.441

Table 5: Comparison of the performance of SVM (Negi and Buitelaar, 2015), LSTM and CNN with the best results reported in two of the related works whose datasets are available. 5 fold cross validation was used. The related works used different kinds of F1 scores.

Train/Test	Precision			Recall			F-1 score		
	SVM	LSTM	CNN	SVM	LSTM	CNN	SVM	LSTM	CNN
Sugg-Forum/Hotel	0.327	**0.425**	0.348	0.156	**0.482**	0.379	0.211	**0.452**	0.363
Sugg-Forum/Electronics	0.109	**0.500**	0.376	0.519	0.532	0.411	0.180	**0.516**	0.393
Sugg-Forum/Travel advice	0.386	**0.52**	0.395	0.212	0.235	**0.531**	0.273	0.323	**0.453**
Travel advice/Hotel	0.147	**0.244**	0.206	0.616	0.616	0.582	0.238	**0.349**	0.304
New Tweets/Microsoft Tweets	0.112	**0.189**	0.164	0.122	0.351	**0.458**	0.117	**0.246**	0.241

Table 7: Cross-domain evaluation: Performance of SVM, LSTM, CNN when trained on new suggestion rich datasets and tested on the existing suggestion datasets. The listed results are for the positive (suggestion) class only.