Inducing Distant Supervision in Suggestion Mining through Part-of-Speech Embeddings

Sapna Negi and Paul Buitelaar
Insight Centre for Data Analytics
National University of Ireland Galway
firstname.lastname@insight-centre.org

Abstract

Mining suggestion expressing sentences from a given text is a less investigated sentence classification task, and therefore lacks hand labeled benchmark datasets. In this work, we propose and evaluate two approaches for distant supervision in suggestion mining. The distant supervision is obtained through a large silver standard dataset, constructed using the text from wikiHow and Wikipedia. Both the approaches use a LSTM based neural network architecture to learn a classification model for suggestion mining, but vary in their method to use the silver standard dataset. The first approach directly trains the classifier using this dataset, while the second approach only learns word embeddings from this dataset. In the second approach, we also learn POS embeddings, which interestingly gives the best classification accuracy.

1 Introduction

Sentences expressing advice, tips, and recommendations can often be found among the opinionated texts like reviews, blogs, tweets, discussions etc (see Table 1). Such sentences can be collectively referred to as suggestions. With the increasing availability of opinionated text, methods for automatic extraction of suggestions can be employed in different applications. The automatic extraction of suggestions from a given text is referred to as suggestion mining.

Although humans can infer suggestions from any given informative text, automatic inferring of suggestions might still be a far fetched task for machines. Therefore, suggestion mining was approached with a focus on detecting the existing sentences which express suggestions in an explicit manner (Negi and Buitelaar, 2015). Studies performed in the past have defined suggestion mining as a sentence classification task, where class prediction has to be made on each sentence of a given opinionated text, classes being suggestion and non suggestion. State of the art opinion mining systems primarily focus on identifying sentiment polarity of the text. Suggestion mining remains a very less explored area as compared to Sentiment Analysis, specially in regard to the recent neural network based approaches for learning feature representation, which is the primary focus of this work.

In a standard use of language, suggestions are expressed with specific grammatical properties, often accompanied with keywords like advice, suggest, recommend etc. However, informal and figurative use of language on the web results in a limited performance of the manually constructed rule based classifiers (Negi et al., 2016). We observe that suggestions can be a complex combination of semantic and syntactic features. Sometimes, the features can also depend on the domain or source of data from where suggestion has to be extracted.
because specific topics are highly likely to appear in suggestions from a given domain. Consider two sentences from hotel reviews, *You must take a long and gloomy corridor, with no decoration, which looks like you are in a basic motel.*, and *You must visit the Ice Bar, 15 euros and includes a drink.* Former is annotated as a non-suggestion and later as a suggestion, however they possess similar grammatical properties. In a statistical approach, given the domain specific training data, a classifier can model that suggestions are more likely to talk about a bar, than a corridor.

Neural networks are proving to be highly effective in learning automatic representation of data which is best suited for the task at hand. The existing manually labeled training datasets available for suggestion mining do not seem to be large enough to learn such representations. The problem alleviates when domain specific datasets are not available, since topics and class distribution of suggestions vary with domains. Suggestion sentences in the hand labeled datasets developed by the previous works are very sparse, ranging from 8% to 27% of the total annotated sentences (Negi et al., 2016). Therefore, we looked for existing sources of suggestion expressing sentences, and discovered that wikiHow (see Figure 1) could be one potential source. In this work, we propose and evaluate two approaches of employing a large silver standard dataset assembled using the text from wikiHow and Wikipedia, which we refer to as the Wiki Suggestion Dataset. An open domain evaluation of the approaches is also performed, where the evaluation datasets cover domains which are different from the gold standard dataset used for training the classifier. The contributions of this work can be summarised as:

1. Development of a large silver standard training dataset using wikiHow and Wikipedia.

2. Improving the classifier performance over the baseline methods, using the proposed approaches for leveraging the silver standard dataset.

3. Experimentation with different variations of the proposed approaches provide interesting findings. Most interesting finding is the improved classification performance in the experiments where the words in the datasets are replaced by their pos tags, and 50 dimensional pos embeddings with a very small vocabulary are employed.

2 Related Work

2.1 Suggestion Mining

The previous approaches for suggestion mining include linguistic rules (Brun and Hagege, 2013; Ramanand et al., 2010), and supervised machine learning with manually determined feature types. The main algorithms used for supervised learning were, HMM and CRF (Wicaksono and Myaeng, 2013), Factorisation Machines (Dong et al., 2013), and Support Vector Machines (Negi and Buitelaar, 2015). These statistical classifiers were mostly trained and evaluated on datasets from a single domain. These studies also provided training datasets, which were all smaller than 8000 sentences, with a highly imbalanced class distribution. Only some of these datasets are publicly available. Suggestion class was the minority class in all of these datasets, with its proportion varying from 8% to 27% of all the sentences in the dataset. Recently, Negi et. al (2016) stressed upon the use of neural networks for suggestion mining, and used pre-trained word embeddings with the gold dataset training datasets. They compared different classifiers, including manually formulated rules, SVM with a variety of manually defined lexical, syntactic, and sentiment features, Convolutional Neural Networks, and LSTM Networks. They used 400 dimensional pre-trained embeddings, which were trained using best performing configurations for the Word2Vec algorithm (Baroni et al., 2014). They performed cross-validation on a number of datasets from different domains, and their results show that LSTM Networks perform consistently better with majority of the datasets, as compared to the other methods. Based on these findings, we also employ a LSTM based classification architecture in this work. To the best of our knowledge, this work is the first to use distant supervision for suggestion mining by means of a large open domain silver standard dataset.

2.2 Learning Continuous Representations of Words

Pennigton et al. (2014) provided GloVE algorithm to train general purpose word embeddings which were shown to outperform other high performance algorithms on various benchmark tasks and datasets. The GloVE outperformed algo-
rithms included the algorithms like *skip grams* and *CBOW* which are two variations of the popular *word2vec model*. Therefore, the pre-trained GloVE embeddings used in (Pennington et al., 2014) are a strong baseline to evaluate the performance of the embeddings learned using the wiki suggestion dataset.

Training task specific embeddings have been proven to be useful for other short text classification tasks like sentiment analysis. Tang et al. (2014) trained sentiment specific word embeddings using supervised learning on a large silver standard sentiment dataset for twitter, which was labeled by means of the emoticons present in the tweets.

3 Datasets

In this work, we use two kinds of datasets, existing hand labeled gold standard datasets as well as the proposed silver standard wiki suggestion dataset. Table 2 lists the statistics of the used datasets.

3.1 Wiki Suggestion Dataset: A Silver Standard

Due to the lack of large training datasets and sparse suggestions in the existing datasets, an obvious intuition is to look for existing sources of text on the web where suggestions are automatically identifiable.

One potential source for obtaining suggestion expressing sentences is web based suggestion forums. An example of such suggestion forum for Microsoft Office 365 can be found on the *uservoice* platform, \(^1\), which is a popular customer feedback management service. However, we observe that text from these forums cannot be directly used, and would require further human annotation. This is because such posts also contain many elaborative and conversational sentences along with the suggestions. WikiHow is an online wiki-style community consisting of an extensive database of how-to guides, which spans across a large variety of topics. Wikihow articles are open domain, and range over factual and non-subjective topics like *Clean Shoe Insoles*, to opinionated topics like *Resolve Conflict Effectively*. Each article always comprises of a *Steps* section which lists the main steps of doing a certain thing (topic of the page). However, what interests us most is that there are optional sections like *Tips*, *Warnings*, and *Community Q and A* in addition to the steps section. The Tips and Warning sections contain short and to-the-point list of suggestions and advice about the topic of the article, and are very much alike the sentences which qualify for the positive instances of a suggestion mining training dataset. In order to obtain an equally large number of negative instances of suggestions, we randomly choose equal number of sentences from Wikipedia. Wikipedia mainly contains factual descriptions, and is therefore very less likely to contain expressions of suggestions. There are three main reasons to consider this dataset as a silver standard. Firstly, we do not perform a manual check on each sentence, and the data might not fully adhere to the annotation guidelines used for the gold standard datasets. Secondly, the negative sample from wikipedia is much cleaner as compared to the ambiguous negative instances in the gold standard datasets. Thirdly, the types of suggestion expressions are likely to be less varied than the gold standard datasets.

**Pre-processing:** We extracted all the items under the tips and warning sections. These items can sometimes be longer than one sentence. We observe that the main tip is mostly provided in the first sentence, and the following sentences contain further details and explanations, which may/may not carry an explicit expression of suggestion. Therefore, we perform automatic sentence splitting of each item and only retain the first sentence as a suggestion instance. Before performing the sentence split, we use regular expressions to re-
move URLs, lists within a single item, and any content within brackets (including the brackets).

3.2 Gold Standard Datasets

As we stated previously, some hand labeled datasets are already available from the previous works. We use the labeled datasets provided by Negi et al. (Negi and Buitelaar, 2015), (Negi et al., 2016), who have performed a detailed annotation study for suggestion mining, and have set guidelines for hand labeling of datasets. The domains which are covered by these datasets are: hotel and electronics reviews, travel discussion forums, suggestion forums for software developers, and twitter. The twitter domain is out of scope for the current work. The hotel review dataset is the largest of all, which we employ as the gold standard training dataset in this work. In all of our experiments, we balance out this training dataset by oversampling the minority class.

The hotel dataset is highly imbalanced, with a very small number of suggestions, and further splitting it into train and test samples would result in even lower number of distinct suggestions in the training sample. Therefore, we prepare a test dataset for hotel reviews, based on the same annotation guidelines as the rest of the datasets. A total of 5023 sentences were annotated by two annotators, out of which 400 were labeled as suggestions by both the annotators. We use a balanced subset of this dataset by only keeping 400 randomly selected non-suggestions, and removing the rest. We also use travel discussion and software suggestion datasets as test datasets, but we use the balanced subsets of the original versions of these datasets. We use a balanced subset of all the datasets because different domains tend to have different distribution of suggestions. In order to evaluate an approach for all the domains, we want to avoid the domain induced class distribution bias in the classifier and therefore use balanced train and test datasets.

4 Methodology

We propose two approaches to employ distant supervision into modeling the classifiers for suggestion mining, using the wiki suggestion dataset. Each approach comprises of different variations.

4.1 Approach 1: Training data for the classifier

In this approach, we induce distant supervision through the wiki dataset by using it as a gold standard dataset, i.e. for training the classifier. Negi et al. (Negi et al., 2016) showed that using pre-trained embeddings is useful for cross domain use of a classifier trained on only a particular domain. We also use the vocabulary and initial weights of the word vectors from the pre-trained GloVE embeddings. Pre-trained GloVE embeddings are freely available for download at Stanford Natural Language Processing group’s web portal. The embedding weights from GloVE are updated each time we train the classifiers; therefore, the words which are not present in the training corpus retain the weights from GloVE. We evaluate all the available dimensions of the pre-trained GloVE vectors, i.e. 200, 100, and 50.

This approach is evaluated with the following variations in the training data:

- **Wiki dataset**: The full wiki suggestion dataset is used for training.

- **Wiki and hotel dataset**: Training is performed in two passes. The classifier is first trained on the entire wiki dataset. The wiki trained model is then further trained on the entire hotel train dataset.

- **Semantic subsample of wiki dataset**: A semantically related subset of wiki dataset is used for training. The subset is obtained in the following steps:
  1. Obtain a bag of all the unique words of the test set.
  2. Obtain a vector of these bag of words by performing an addition of the corresponding word embeddings.
  3. Obtain a vector for each sentence in the wiki dataset by using the same method as above.
  4. Calculate the similarity score of the test set vector with each of the wiki sentence vector.
  5. Choose the top 1% similar sentences from both the classes of the wiki dataset.

This results in a balanced subset of about 13,500 sentences. Only 100 dimensional pre-trained embeddings are used in this experiment, since
they tend to perform best in the previous two settings.

- **Wiki subsample and hotel dataset:** Same as the variation #2, except that the semantic subsample of wiki dataset is employed in place of the full wiki dataset.

**Baseline:** The baseline for this approach is to use an actual gold standard dataset, which is hotel train dataset.

### 4.2 Approach 2: Training the embeddings

In this approach, wiki dataset is used to induce distant supervision in the classifier model only through the pre-trained word embeddings. We first train word embeddings using the wiki suggestion dataset, and then replace the pre-trained GloVE vectors in the previous approach with these vectors, and train the network only with the gold standard dataset. Following are the different methods we employed to learn embeddings.

**Wiki Suggestion Embeddings (WiSE):** We train a classifier using the full wiki suggestion dataset, without using any pre-trained word embeddings for the initial weights. Word weights are then extracted from the embedding layer at the end of the training. The classification architecture remains same as that of the approach 1. We refer to these embeddings as the Wiki Suggestion Embeddings (WiSE). These embeddings are different from the standard co-occurrence based embeddings (GloVE, Word2vec etc.) since they are learned using a class prediction objective. We investigate two variations of the WiSE embeddings:

- **WiSE\textsubscript{w}**: These are the embeddings for words. The vocabulary comprises of words which are present in the wiki suggestion dataset, and is therefore smaller than that of the GloVE vectors. These embeddings are trained for 100 dimensions.

- **WiSE\textsubscript{p}**: These are the embeddings for Part of Speech tags. These embeddings are very light weight, with 50 dimensions and a vocabulary of size 34. These embeddings are learned by replacing all the words in the wiki corpus with their POS tags.

Specific syntactical and grammatical properties are strong features for suggestions, and suggestions across domains tend to exhibit similar syntactic properties (Negi et al., 2016). Earlier works on suggestion mining also show that POS n-gram and POS pattern based features are useful with SVM classifiers (Negi and Buitelaar, 2015). Therefore, using word vectors which strongly encode syntactic properties may be a promising direction for open domain training of suggestion mining classifiers. It has been shown earlier that the smaller dimensions of SOTA word embeddings like GloVe tend to capture syntactic properties of the words (Plank et al., 2016) and perform well for syntactic analogy tasks. We therefore propose to learn embeddings for part of speech tags, as an alternative to the 50 dimensional word embeddings.

**Glove-Wiki\textsubscript{p}:** These are the POS embeddings learned using the GloVe algorithm, but trained on the wiki suggestion dataset.

**Concatenation of Word and POS embeddings:** These are the embeddings obtained by concatenating the best performing word embeddings with the best performing POS embeddings.

**Baseline:** The baseline for these embeddings are the pre-trained GloVE embeddings of the same di-
5 Network Architecture and Hyper-parameters

We evaluated a few popular configurations and hyper-parameter values for our experiments. These configurations were evaluated on the baseline approach, i.e. using hotel train dataset with all three dimensions of the GloVE embeddings. The options include:

- Activation functions: tanh, relu, softsign
- Dropout (0.2), l2 regularisation (0.02).
- Optimisation: SGD, Adam.
- 1, 2, 3 hidden layers.
- LSTM hidden layers, fully connected dense hidden layers.
- 10, 15, 20 units in the last hidden layer.
- 5, 8, 10 epochs.

![Figure 2: Architecture for the LSTM based neural network classifier](image)

We chose the configurations which perform best for at-least two out of the three test datasets. The final architecture comprises of 2 lstm hidden layer of 50, 20 units respectively. The input layer comprises of LSTM units, and a softmax activation is applied to the outermost dense layer of size 2. Tanh activation is used at each layer, L2 regulariser is used with the first three LSTM layers, Adam is used for optimisation, and the training was performed with 5 epochs. Keras (theano backend), Gensim, and scikit learn libraries are primarily used for the implementation.

![Figure 3: Training sample size of wiki dataset vs. precision, recall and F1 score on the travel dataset](image)

6 Results and Analysis

Tables 3, 4 show the Precision, Recall, and F1 score of the suggestion class on different test datasets, for approaches 1 and 2 respectively. For approach 1, using wiki training dataset outperforms all other variations of training datasets for all three test datasets. However, the performance improvement for the domain whose training dataset is available (hotel) is small as compared to the other domains.

For approach 2, results demonstrate that the use of pre-trained POS embeddings yields better F-scores as compared to the baseline of using pre-trained glove embeddings. Based on the current experiments, approach 2 appears to be better for using wiki suggestion dataset. The results are very interesting considering the fact that the training vocabulary for approach 2 (only PoS tags) is much smaller than the standard word embeddings.

Large wiki dataset vs small gold standard dataset: Travel test data showed the largest improvement in F1 score on using wiki dataset as training dataset. Figure 3 shows the change in precision, recall, and F1 score with the increasing size of wiki training dataset in approach 1, which shows that the size of the wiki dataset is the reason for its better performance over the gold standard. The results improve with the increasing size of the dataset.

Software test data showed the largest improvement in F1 score on using WiSEp embeddings. Figure 4 shows the change in precision, recall, and F1 score of the software test dataset when the
### Table 3: Results from different dataset settings using gold standard hotel and Wiki Suggestion dataset to directly train a classifier. Precision, Recall, and F measure are provided for the positive class, i.e. the suggestion class.

| Train data          | Embedding | Hotel Test | Travel | Software |
|---------------------|-----------|------------|--------|----------|
|                     |           | P  | R  | F1 | P  | R  | F1 | P  | R  | F1 |
| Hotel (Baseline)    | Glove 200 | 0.99| 0.59| 0.74| 0.78| 0.26| 0.40| 0.70| 0.28| 0.40|
|                     | Glove 100 | 0.95| 0.71| 0.81| 0.71| 0.35| 0.47| 0.64| 0.49| 0.55|
|                     | Glove 50  | 0.97| 0.57| 0.72| 0.72| 0.28| 0.40| 0.69| 0.39| 0.50|
| Wiki                | Glove 200 | 0.75| 0.90| **0.82**| 0.57| 0.70| **0.63**| 0.55| 0.75| 0.64|
|                     | Glove 100 | 0.76| 0.88| **0.82**| 0.56| 0.69| 0.62| 0.55| 0.76| 0.64|
|                     | Glove 50  | 0.76| 0.88| **0.82**| 0.56| 0.70| 0.62| 0.56| 0.79| **0.66**|
| Wiki + Hotel        | Glove 200 | 1.00| 0.29| 0.45| 0.78| 0.06| 0.12| 0.78| 0.07| 0.13|
|                     | Glove 100 | 1.00| 0.38| 0.55| 0.83| 0.14| 0.23| 0.71| 0.21| 0.32|
|                     | Glove 50  | 1.00| 0.38| 0.55| 0.81| 0.18| 0.29| 0.73| 0.33| 0.45|
| Wiki subsample      | Glove 100 | 0.99| 0.19| 0.32| 0.70| 0.14| 0.23| 0.52| 0.74| 0.61|
| Wiki subsample +    | Glove 100 | 0.99| 0.39| 0.74| 0.82| 0.14| 0.24| 0.64| 0.12| 0.21|

### Table 4: Classification results of using different embeddings with hotel data as the training dataset. Precision, Recall, and F1 score for the suggestion class are presented.

| Embedding          | Dimensions, Vocabulary | Hotel Test | Travel | Software |
|--------------------|------------------------|------------|--------|----------|
|                    |                        | P  | R  | F1 | P  | R  | F1 | P  | R  | F1 |
| Glove (Baseline)   | 100, 6B               | 0.95| 0.71| 0.81| 0.71| 0.35| 0.47| 0.64| 0.49| 0.55|
| WiSE<sub>wp</sub>  | 100, 339,240          | 0.26| 0.51| 0.34| 0.25| 0.50| 0.33| 0.25| 0.50| 0.33|
|                    |                        | Part of Speech Embeddings |          |
| Glove (Baseline)   | 50, 6B                 | 0.97| 0.57| 0.72| 0.72| 0.28| 0.40| 0.69| 0.39| 0.50|
| Glove-Wiki<sub>wp</sub> | 50, 34           | 0.80| 0.83| 0.82| 0.57| 0.71| **0.63**| 0.57| 0.90| 0.70|
| WiSE<sub>wp</sub>  | 50, 34                 | 0.82| 0.84| 0.83| 0.57| 0.69| 0.62| 0.38| 0.90| **0.71**|
| Glove + WiSE<sub>wp</sub> Concat | 130, 6B * 34 | 0.90| 0.80| **0.85**| 0.62| 0.57| 0.39| 0.61| 0.81| 0.70|
Figure 4: Training sample size for WiSE vs. precision, recall and F1 score on the software dataset.

| Test Data | P   | R   | F   |
|-----------|-----|-----|-----|
| Hotel     | 0.58| 0.50| 0.55|
| Travel    | 0.52| 0.55| 0.53|
| Software  | 0.53| 0.59| 0.56|

Table 5: Classification results when POS version of wiki dataset is directly used to learn the classification model, which also includes learning WiSE_p.

POS embeddings are learned on the increasing size of wiki dataset in approach 2, which shows that the large size of dataset doesn’t contribute much in this case. This is in line with our observation that the suggestion expressions do not vary much in this dataset. Table 5 shows the results when we use only the pos version of wiki dataset to train the classifier in approach 2, but the results are much lower compared to using the gold standard (Table 4). This hints that the model which uses POS embeddings can model syntactic variations in suggestions with a much smaller training dataset as compared to using word embeddings.

**Wiki dataset insufficient for learning word embeddings:** Table 4 show that WiSE_w performed poorly in the approach 2. Table 6 show that the a large portion of test dataset vocabulary is not present in the wiki dataset, therefore the vocabulary of the WiSE_w embeddings is insufficient for learning task specific word embeddings.

### Conclusion

Our experiments have revealed several interesting observations. Most interesting results are the high performance when using pre-trained POS embeddings with part of speech versions of the datasets, which is an extremely lightweight model. This provides a ground for evaluating this approach on other sentence level tasks.

The two approaches to use wikihow dataset perform better than the baseline, the POS embedding approach in particular outperforms all other variations. This proves that syntax and grammar are predominant features in determining the suggestion class. However, it is also evident that in order to achieve higher classification accuracy, domain data would still be required. Last but not least, to the best of our knowledge this is the first attempt to use wikiHow for a text classification task.

In future we would like to investigate different neural network architectures to compose the word and POS embeddings against the simple concatenation method used in this work.

### Acknowledgments

This publication has emanated from research conducted with the financial support of Science Foundation Ireland (SFI) under Grant Number SFI/12/RC/2289 (Insight).

### References

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)*. Association for Computational Linguistics, Baltimore, Maryland, pages 238-247. [http://www.aclweb.org/anthology/P14-1023](http://www.aclweb.org/anthology/P14-1023).

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*.

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.

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. Association for Computational Linguistics, Berlin, Germany, pages 170–178. http://anthology.aclweb.org/S16-2022.

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, Lisbon, Portugal, pages 2159–2167. http://aclweb.org/anthology/D15-1258.

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. http://www.aclweb.org/anthology/D14-1162.

Barbara Plank, Anders Søgaard, and Yoav Goldberg. 2016. Multilingual part-of-speech tagging with bidirectional long short-term memory models and auxiliary loss. arXiv preprint arXiv:1604.05529.

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. Association for Computational Linguistics, Los Angeles, CA, pages 54–61. http://www.aclweb.org/anthology/W10-0207.

Duyu Tang, Furu Wei, Nan Yang, Ming Zhou, Ting Liu, and Bing Qin. 2014. Learning sentiment-specific word embedding for twitter sentiment classification. In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Association for Computational Linguistics, Baltimore, Maryland, pages 1555–1565. http://www.aclweb.org/anthology/P14-1146.

Alfan Farizki Wicaksono and Sung-Hyun Myaeng. 2013. Automatic extraction of advice-revealing sentences for advice mining from online forums. In K-CAP. ACM, pages 97–104.