AKTSKI at SemEval-2016 Task 5: Aspect Based Sentiment Analysis for Consumer Reviews

Shubham Pateria and **Prafulla Kumar Choubey**
System Software Division
Samsung R and D Institute India Bangalore Pvt Ltd
Bangalore, Karanataka, India -560037
s.pateria, prafulla.ch@samsung.com

Abstract

This paper describes the polarity classification system designed for participation in SemEval-2016 Task 5 - ABSA. The aim is to determine the sentiment polarity expressed towards certain aspect within a consumer review. Our system is based on supervised learning using Support Vector Machine (SVM). We use standard features for basic classification model. On top this, we include rules to check precedent polarity sequence. This approach is experimental.

1 Introduction

In the consumer-focused markets today, understanding opinions expressed on the online platforms or review portals is of key essence for the businesses. Statistical or Machine learning methods and Natural Language Processing are now being widely applied to extract important information or patterns from the opinion data. A review statement may have a mix of sentiments towards different aspects. For e.g., consider *the food and ambiance at xyz hotel were extraordinary, as expected. However, the waiters seemed rude*. Here, the main entity of review is a 'hotel'. Henceforth, we will refer to such main entity as *global item*. However, there is no definite overall sentiment expressed towards the *global item*. Different sentiments are expressed towards *food* and *ambiance* aspects (extraordinary: positive) and towards the aspect of service (*waiters*, rude: negative). Thus, it is important to approach sentiment detection as an aspect-based problem.

The SemEval-2016 Task 5 - Aspect Based Sentiment Analysis (ABSA) focuses on this problem (Pontiki et al., 2016). This task is a continuation from SemEval-2015 ABSA task (Pontiki et al., 2015). The task was organized across different domains and languages. We participated in Restaurant domain in English language. The focus of our system is polarity detection and not aspect extraction. Thus, we use dataset in which aspects are already known.

To develop our system, we have used standard features for basic model and also rules to check effect of precedent-polarity sequence pattern on polarity to be predicted. We focus on experimenting with sequence pattern. The system is described in Section 3. Pre-processing is described in Sub-section 3.1. selected features are discussed in Sub-section 3.2 and sequence pattern discussed in Sub-section 3.3. In section 4, we discuss the analysis and evaluation results for our system.

2 Related Work

Aspect-based sentiment analysis has been a subject of some interesting works so far. (McAuley et al., 2012) employ topic modeling paradigm to address this problem. Deep Learning has also been explored in this area, such as by (Wang and Liu, 2015). They used Convolutional Neural Network for aspect-based analysis of SemEval-2015 ABSA data and reported performance comparable to top systems of the 2015 task. Previously, the system by (Kiritchenko et al., 2014) achieved the best performance in Polarity Detection task in SemEval-2014. They used various innovative linguistic features, publicly available sentiment lexicon corpora and automatically generated polarity lexicons. In

Proceedings of SemEval-2016, pages 330–336,
San Diego, California, June 16-17, 2016.

Semeval-2015, SENTIUE system by (Saias, 2015) provided remarkable results. They used wide range of features such as Bag-of-Words, negation words, bigram after negation, polarity inversion, polarized terms in last 5 tokens, publicly available lexicons etc. They used MALLET[1] with Maximum Entropy classifier.

3 Classification System

Our system uses Support Vector Machine (SVM) with Radial Basis Function (RBF) kernel as classifier. The scikit-learn SVM implementation has been used (Pedregosa et al., 2011). This classifier is trained using dataset provided by task organizers. This dataset consists of several reviews, each with a unique review ID (rID). Each review consists of several sentences. A sentence may have single or multiple aspects. Sentences under same rID express sentiment towards the same *global item*. In our case, the *global item* is some *restaurant*. The data are parsed into following format:
{Review(rID) {Sentence1 aspect1:(target, category, polarity, from, to)}, {Sentence2 aspect1:(target, category, polarity, from, to)}, ..., {SentenceN aspect1:(target, category, polarity, from to) ... aspectM:(target, category, polarity, from, to)} }.

Here, Review(rID) is just one instance out of several such reviews. (target, category, polarity, from, to) are values belonging to an aspect of a sentence. Polarity values are *positive*, *negative* or *neutral*. SentenceN is an example of a sentence which contains multiple aspects. The test data are also parsed in the same format except that polarity values are not provided. Henceforth, (target, category, polarity, from, to) will be referred to as (tar, cat, p, f, t) for simplicity.

To develop our system, we have used NLTK package (Loper and Bird, 2002) in Python with resources such as WordNet package[2], SentiWordNet, Bing Liu's opinion lexicon and MPQA subjectivity lexicon[3].

3.1 Pre-processing

Consider following sentence,
Chow fun was dry; pork shu mai was more than usually greasy and had to share a table with loud and rude family[4]

For this sentence, we have following (tar, cat, p, f, t) values:
target="Chow fun", category="FOOD#QUALITY", polarity="negative", from="0", to="8"; target="pork shu mai", category="FOOD#QUALITY", polarity="negative", from="18", to="30"; target="NULL", category="RESTAURANT#MISCELLANEOUS", polarity="negative", from="0", to="0"

Here, from-to values provide the location of *tar* within the sentence.

Based on the observation made on the provided dataset, we hypothesize that only the terms related to *tar* affect the aspect-polarity *p*. In the example above, only "more than usually greasy" is relevant for "pork shu mai". Thus, first we decompose any {SentenceX aspect1:(tar, cat, p, f, t) ... aspectM:(tar, cat, p, f, t)} into {SubSent1 aspect1:(tar, cat, p, f, t), ..., SubSentM aspectM:(tar, cat, p, f, t)}, where M is greater than or equal to 1 and SentenceX is any sentence with aspect values assigned to it.

For decomposition, we first use Stanford Dependency Parser (de Marneffe et al., 2006) to obtain a dependency graph of SentenceX. Using the obtained graph, we choose terms in SentenceX which are more closely related to *tar* terms. For e.g., in *the staff acted like we were imposing on them and they were very rude*, the underlined terms are related in the dependency graph. Here, *tar* is 'staff'. SubSent for any *tar* can be formed using only such related terms.

SubSent formation is not straightforward when *tar* is NULL. We use self-generated *tar* values in such cases. Our intuition is that the terms express-

[1]MAchine Learning for LanguagE Toolkit (McCallum and Kachites, 2002)

[2]Princeton University "About WordNet." WordNet. Princeton University. 2010. <http://wordnet.princeton.edu>

[3]Bing Liu's lexicon (Liu et al., 2005; Liu, 2012), SentiWordNet (Esuli and Sebastiani, 2010), MPQA subjectivity clues (Wiebe et al., 2005). Bing Liu's lexicons and SentiWordnet are available as part of NLTK package. Bing Liu's lexicons and MPQA are binary, i.e., they simply classify words or terms as positive or negative. SentiWordNet provides a range of positive and negative scores for terms.

[4]This sentence, and all sentences henceforth, are taken from training dataset.

ing sentiment should be related to a noun or pronoun subject (for instance, "loud" and "rude" related to *family*). Thus, after eliminating all SubSent for non-NULL *tar*, sentiment terms in the remaining sentence are identified by looking-up terms in the lexicon corpora. Then, a noun or pronoun related (in dependency graph) to identified terms is considered as *tar*. Since the *global item* is restaurant, if 'food', 'drinks', 'service', 'waiter', 'price', 'staff' or 'ambiance' are present, they are preferably considered as *tar*. Also, 'they', 'she' and 'he' are frequently used to refer to service staff in the provided dataset. Hence, these terms are also preferred as *tar*.

After decomposing, we filter-out stop words selected from NLTK's stop word list. Numbers and symbols (except '!') are also filtered-out using regular expression.

3.2 Features

Following basic features have been used in our model:

1. **Sentiment lexicons** - Separate features for Bing Liu's (binary), MPQA (binary) and SentiWordNet (range of scores).
   Presence of negation terms : The scores of sentiment lexicons are modified according to negation (e.g., 'not', 'didn't', 'don't' etc.). Bing Liu and MPQA features are simply reversed (pos $\rightarrow$ neg, neg $\rightarrow$ pos). For SentiWordnet features, negation is made in proportion to the scores. For e.g., a word like 'extraordinary' having higher positive score is less affected by negation compared to a word like 'good' having lower positive score. We use a simple scheme for score modification: pos = pos + $\frac{(neg-pos)}{2}$ and neg = neg + $\frac{(pos-neg)}{2}$. Here, pos and neg are positive and negative lexicon scores of a term, receptively. A significant work on negation problem has been done by (Zhu et al., 2014). They provide several methods to perform shifting of sentiment scores.

2. **Uni-grams and Bi-grams** extracted from each SubSent.

3. **Self-generated list of neutral terms** - Based on observation made on provided training dataset, we found that following terms frequently occur in "neutral" polarity SubSent(s): 'average', 'normal', 'simple', 'okay', 'ok', 'not great', 'nothing great', 'moderate', 'typical', 'alright', 'fair', 'mediocre', 'just', fine', 'not too good', 'good enough'
   These terms and phrases do not necessarily fall in positive or negative category of lexicon features. Hence, these are used as separate unigrams. Terms in a phrase like 'not too good' are concatenated as 'not0too0good'.

4. **Punctuation** like '!'. In the training dataset, this punctuation mostly co-occurs with positive polarity. Hence, the occurrence of the punctuation is checked.

5. **Keywords associated with specific aspect category** - There are a total of 12 aspect-categories (*cat*) such as FOOD#QUALITY, FOOD#PRICES, RESTAURANT#GENERAL, SERVICE#GENERAL etc. in the provided dataset. For a specific *cat*, there could be keywords which, when co-occurring with the *cat*, express some sentiment. For e.g., high and low are generic terms but for FOOD#PRICES they can indicate a polar sentiment. We divide the dataset into 12 documents, one for each *cat*. Then, we identify keywords based on Term Frequency - Inverse Document Frequency (TF-IDF) scores. The frequently occurring terms are added to a keyword list. Frequency thresholds of min:0.3 & max:0.8 are used. Total 12 keyword lists are obtained, one for each *cat*. Then, for each {SubSent aspect:(tar, cat, p, f, t)}, we check for presence of keywords corresponding to *cat* in SubSent. If found, the keywords are used as new uni-gram features.

These are the features used for basic classification model. In the next sub-section, we describe inclusion of sequence pattern.

3.3 Using precedent polarity sequence (experimental)

Following observations are made on the provided training dataset:
1. In majority of the cases, the sentences under the

same rID exhibit similar sentiment. In other words, polarity values $\{p_1, p_2, ..., p_N\}$, under same rID, are equal. Henceforth, we will refer to this as *Flow*.

2. There are sentences where the polarity values change, i.e., p_i is not equal to p_{i-1}, under same rID. Henceforth, we will refer to this as *Trans* (transition). *Trans* instances may be identified by explicit contrast terms present around *tar*. The common contrast terms found in the dataset are:
'but', 'however', 'though', 'tho', 'although', 'yet', 'except'

For instance, *The decor is right tho...but they REALLY need to clean that vent in the ceiling...its quite un-appetizing, and kills your effort to make this place look sleek and modern*
target="place" polarity="negative"; target="decor" polarity="positive"; target="vent" polarity="negative"
However, this does not imply that a contrast term is always present when *Trans* happens.

Exploiting the '*Flow* or *Trans*' patterns can help address ambiguity. This is the main reason for including sequence pattern. Consider following sentence:
The manager came to the table and said we can do what we want, so we paid for what we did enjoy, the drinks and appetizers.
For a classifier, the sentiment expressed towards 'manager' may be ambiguous. Our basic model classifies this as *neutral*, while the true polarity is *negative*. However, if we take previous sentence in consideration - *The level of rudeness was preposterous* - the state of mind of the reviewer becomes more clear.

Based on this observation, we hypothesize that, under same review (rID), precedent polarity outcome affects current polarity outcome, either by *Flow* or *Trans*, given certain conditions. (Vanzo et al., 2014) propose a context-based model for sentiment analysis of tweets, on similar lines. They use sequence of tweets to build Conversational context, hashtags to build Topical context and also use Markovian approach.

We describe our methods to account for *Flow* or *Trans* here.

Method1: We use new set of features instead of basic feature-set discussed in sub-section 3.2. First, we generate the features representing conditions for *Flow* or *Trans*. We use two conditions for our model - contrast keywords and sentiment keywords - present in a SubSent. The training dataset is divided into 3 sub-sets according to polarity labels. Then, we search for sentiment terms belonging to one of the lexicon corpora, sentiment terms with negation terms (bi-grams and tri-grams) and terms belonging to our neutral word list. A new dictionary *D* is created with these terms. Moreover, TF-IDF based selection is performed on the 3 sub-sets (or documents). Frequency thresholds are min:0.3 & max:0.8. This ensures inclusion of any frequent keywords which are not already a member of *D*. Then, for a $SubSent_i$, feature set $X_i = \{posD, negD, neutD, cont\}_i$ is generated. Here,
posD: terms in a SubSent belonging to positive section of *D*; *negD*: terms in a SubSent belonging to negative section of *D*; *neutD*: terms in a SubSent belonging to neutral section of *D*; *cont*: contrast terms in SubSent.
Separate sentiment classes have been used here to let the classifier learn how strongly a SubSent is inclined towards any particular sentiment type. The classifier should learn that if such inclination is strong, then ambiguity is low. So, effect of previous SubSent should also be low.

New input feature-set corresponding to $SubSent_i$ is **X(i)** = $\{X_i, X_{i-1}, X_{i-2}\}$, plus, selective features from sub-section 3.2. For initial two SubSent(s) under a rID, X_{i-2} or both X_{i-1} and X_{i-2} are empty. We do not use n-gram and neutral word features because terms are now selected from *D*. Punctuation is ignored since its effect is minimal (Table 1). *cat* specific keywords are included because they are extracted using different document-types. Lexicon scores are also included to capture sentiment strength. The same SVM-RBF classifier is then trained with **X** to predict polarities. For test data, same dictionary *D* is used to generate new features.

Method2: This method is along the lines of auto-regression[5]. However, polarity sequence is not a strict time-series. Hence, we devise our mathematical model with necessary considerations. A first set of predicted polarities $P_1 = \{p_{11}, p_{12}, ..., p_{1k}\}$ are obtained using SVM-RBF with all of the basic

[5]<http://paulbourke.net/miscellaneous/ar/>

features from sub-section 3.2. Polarities are mapped as {*positive, negative, neutral*} → {1,-1,0}. The aim is to obtain final predictions, P_2 = {p_{21}, p_{22}, ..., p_{2k}}. Feature-set X_i = {$posD_i$, $negD_i$, $neutD_i$, $cont_i$} for $SubSent_i$ of test data is obtained using *D*. However, we do not predict using these features. The *Flow* or *Trans* effect is directly calculated using P_1 values. For each $SubSent_i$, we define following values:

s_i^p : positive vote. This is initialized by *0*, then incremented by *+1* for first term found in $posD_i$ and by *+0.5* for every next term in $posD_i$,

s_i^n : negative vote. This is initialized by *0*, then incremented by *-1* for first term found in $negD_i$ and by *-0.5* for every next term in $negD_i$,

s_i^o : neutral vote. This is initialized by *0.4* (s_{imin}^o), then incremented by (1-s_i^o)/4 for every term found in $neutD_i$, keeping the value below 1.

c_i : contrast vote. This is initialized by *+1*; assigned c_i = 2, if $cont_i$ is not empty,

w_i : aggregate voting weight. This is calculated as,

w_i = $|p_{1i}|(|(p_{1i}+1)/2|(2s_i^p + s_i^n) + |(p_{1i}-1)/2|(s_i^p + 2s_i^n) + s_{imin}^o) + ||p_{1i}|-1|s_i^o$,

Since, a SubSent must express some sentiment, we assume a basic neutral characteristic in each SubSent. Hence, s_{imin}^o is added.

We define a function *g(w,p)* = |w|(p + ||p|-1|). Then, using these parameters we calculate a weighted effect, $\hat{p}$(i), for polarity as,

$E_i = g(w_{i-1},p_{1,i-1}) + \sum_{r=l}^{r=i-1}(c_r/c_{r-1})g(w_{r-1},p_{1,r-1})$,

$E_{i(avg)} = E_i/(1+\sum_{r=l}^{r=i-1}(c_r/c_{r-1}))$,

$\hat{p}(i) = g(w_i,p_{1i}) + E_{i(avg)}/2c_i$.

The increment and assignment values have been chosen after experimenting with different values. Also, for our model, *l = i-2* works best. Effect value E_i captures the effect of precedent polarities. The effect of a polarity value $p_{1,i-2}$ should be amplified with respect to $p_{1,i-1}$ if $p_{1,i-1}$ came by contrast and reduced if $p_{1,i-2}$ itself came by contrast. Hence, $p_{1,i-2}$ is multiplied with c_{i-1}/c_{i-2}. Finally, the average effect $E_{i(avg)}$ should be reduced if current $SubSent_i$ has explicit contrast terms. Hence, the division by $2c_i$. Then, if $\hat{p}(i)<0$, p_{2i} = -1; if $\hat{p}(i)>1$, p_{2i} = 1; otherwise, p_{2i} = 0.

These equations are tuned based on observations made on training data. More generic and robust equations need to be formed. This needs further investigation.

Features	**Accuracy**
n-grams only	0.61 (+/- 0.04)
lexicons-with-negation (lx) only	0.64 (+/- 0.06)
n-grams + lx	0.69 (+/- 0.05)
n-grams + lx + neutral terms (nt)	0.71 (+/- 0.04)
n-grams + lx + nt + punctuation	0.71 (+/- 0.05)
n-grams + lx + nt + keywords (kw)	0.75 (+/- 0.04)
Method1 + n-grams + lx + nt + kw	0.79 (+/- 0.04)
Method1 + lx + kw	0.80 (+/- 0.04)
Method2	0.82 (+/- 0.04)

Table 1: Model performance on training dataset.

4 Analysis and Evaluation Results

4.1 Analysis using training data

The analysis of our system on training data is provided in Table 1. SVM-RBF with parameters : [C=100, kernel='rbf', gamma = 0.001] is used (same for evaluation/test). Parameters are obtained using grid search. The accuracy scores are obtained using 10-fold cross-validation from scikit-learn (Pedregosa et al., 2011). N-grams obtained using dependency relation with aspect-target are base features. Lexicons are essential to capture sentiment types and scores. However, we found that there were some terms occurring in neutral sentences which were not listed in lexicon corpora. Hence, we generated our own list of neutral words. Punctuation (!) has negligible effect on the performance. Including aspect-category keywords improves accuracy. As discussed earlier, keywords are required to include terms that express some opinion specific to a category. These are the only basic features used. On top of this, we include polarity sequence pattern. It can be seen that Method2 provides better result than Method1 by a slight margin only. Method2 may not be necessarily better, but we prefer using it. It theoretically permits using more than one precedent polarities in the sequence, if required, without involving complex features; only the summation series needs to be expanded as we go along a polarity sequence. Method2 is used in final Evaluation model.

Due to time constraint, we focus more on inclusion of polarity sequence pattern instead of engineering better features or classifier ensemble.

System	Accuracy	Ratio	Rank
AKTSKI	0.717	616/859	24
Highest1 : XRCE	0.881	757/859	1
Highest2 : IIT-TUDA	0.867	745/859	2
Baseline	0.764	657/859	21

Table 2: Evaluation results. Ratio is no. of correct predictions/total no. of aspects. Accuracy values are on scale of 0 to 1.

Score	System1	System2	System3
Accuracy	0.668	0.702	0.717
Precision	0.480	0.510	0.500
Recall	0.482	0.500	0.506

Table 3: Comparative performance. System1: without sequence pattern, System2: using Method1, System3: Using Method2. Accuracy values are on scale of 0 to 1.

4.2 Evaluation result

The result of evaluation is provided in Table 2. There were 676 sentences in the evaluation (test) data and 859 instances of aspect values (tar, cat, f, t). The polarity values had to be predicted. The system predictions were evaluated against gold labels by the organizers. There were total 30 submissions in polarity detection task for Restaurant domain and English language. This included multiple submissions from single teams as well. Relative performance of our system was poor. This may be attributed to less effort invested on improving classifier model (using ensembles, or otherwise) or on using more robust features. We also suspect that {posD, negD, neutD, cont} features may be biased towards training data as the keyword dictionary *D* was generated on the full training dataset before evaluation. Moreover, Method2 is tuned using training data and expected to perform weaker on unseen datasets.

4.3 Further evaluation on gold-labeled data

We did further evaluation of our system after release of gold-labeled test data. This was aimed at checking the effect of using sequence pattern. The results are provided in Table 3. The accuracy obtained against gold labels without using sequence pattern was 0.668. By using Method1, the accuracy increased to 0.702. With Method2, the accuracy obtained was 0.717. These are small increments. Also, the method has obvious caveats as mentioned above. So, the usage of sequence pattern needs to be improved by more research.

5 Conclusion

We submitted unconstrained system for sentiment polarity detection. The system was unconstrained in the sense that it used several external resources for feature generation. Apart from standard features, we experimented with polarity sequence pattern. This approach provides slight improvement in prediction accuracy as checked on evaluation data. However, for any serious purpose, this approach requires deeper investigation. Our next step would be to devise more robust feature-extraction to handle polarity sequence patterns. Moreover, this approach needs to be tested on broader datasets. We will also explore using sequence pattern with multiclass Platt Scaling (Zadrozny and Elkan, 2002) or ensemble models to check performance.

References

Marie-Catherine de Marneffe, Bill MacCartney, and Christopher D. Manning. 2006. Generating typed dependency parses from phrase structure parses. In *Proceedings of the fifth international conference on Language Resources and Evaluation (LREC)*.

Stefano Baccianella Andrea Esuli and Fabrizio Sebastiani. 2010. Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In *Proceedings of the 9th international conference on Language Resources and Evaluation (LREC)*.

Svetlana Kiritchenko, Xiaodan Zhu, Colin Cherry, and Saif Mohammad. 2014. Nrc-canada-2014: Detecting aspects and sentiment in customer reviews. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 437 – 442. Association for Computational Linguistics.

Bing Liu, Minqing Hu, and Junsheng Cheng. 2005. Opinion observer: Analyzing and comparing opinions on the web. In *In Proceedings of the 14th International World Wide Web conference*.

Bing Liu. 2012. *Sentiment Analysis and Opinion Mining*. Synthesis Lectures on Human Language Technologies.

Edward Loper and Steven Bird. 2002. Nltk: The natural language toolkit. In *Proceedings of the ACL Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational*

Linguistics. Philadelphia: Association for Computational Linguistics. Association for Computational Linguistics.

Julian McAuley, Jure Leskovec, and Dan Jurafsky. 2012. Learning attitudes and attributes from multi-aspect reviews. In *Proceedings of the 12th IEEE International Conference on Data Mining*, Brussels, Belgium. IEEE Computer Society.

McCallum and Andrew Kachites. 2002. *MALLET: A Machine Learning for Language Toolkit.*

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825 – 2830.

Maria Pontiki, Dimitrios Galanis, Haris Papageorgiou, Ion Androutsopoulos, and Suresh Manandhar. 2015. SemEval-2015 task 12: Aspect based sentiment analysis. In *Proceedings of the 9th International Workshop on Semantic Evaluation*, SemEval '15, pages 486 – 495. Association for Computational Linguistics.

Maria Pontiki, Dimitrios Galanis, Haris Papageorgiou, Ion Androutsopoulos, Suresh Manandhar, Mohammad AL-Smadi, Mahmoud Al-Ayyoub, Yanyan Zhao, Bing Qin, Orphée De Clercq, Vêronique Hoste, Marianna Apidianaki, Xavier Tannier, Natalia Loukachevitch, Evgeny Kotelnikov, Nuria Bel, Salud Maria Jimênez-Zafra, and Gülşen Eryiğit. 2016. SemEval-2016 task 5: Aspect based sentiment analysis. In *Proceedings of the 10th International Workshop on Semantic Evaluation*, SemEval '16, San Diego, California, June. Association for Computational Linguistics.

Josê Saias. 2015. Sentiue: Target and aspect based sentiment analysis in semeval-2015 task 12. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 767 – 771.

Andrea Vanzo, Danilo Croce, and Roberto Basili. 2014. A context-based model for sentiment analysis in twitter. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics*, pages 2345–2354, Dublin, Ireland, August. Association for Computational Linguistics.

Bo Wang and Min Liu. 2015. *Deep Learning For Aspect-Based Sentiment Analysis*. Stanford University report, https://cs224d.stanford.edu/reports/WangBo.pdf.

Janyce Wiebe, Theresa Wilson, and Claire Cardie. 2005. *Annotating expressions of opinions and emotions in language*. Language Resources and Evaluation.

Bianca Zadrozny and Charles Elkan. 2002. Transforming classifier scores into accurate multiclass probability estimates. In *Proceedings of the 8th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, Alberta, Canada.

Xiaodan Zhu, Hongyu Guo, Saif Mohammad, and Svetlana Kiritchenko. 2014. An empirical study on the effect of negation words on sentiment. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, pages 304 – 313. Association for Computational Linguistics.