

SIEL: Aspect Based Sentiment Analysis in Reviews

Satarupa Guha, Aditya Joshi, Vasudeva Varma

Search and Information Extraction Lab

International Institute of Information Technology, Hyderabad

Gachibowli, Hyderabad, Telengana, India

{satarupa.guha, aditya.joshi}@research.iiit.ac.in
vv@iiit.ac.in

Abstract

Following the footsteps of SemEval-2014 Task 4 (Pontiki et al., 2014), SemEval-2015 too had a task dedicated to aspect-level sentiment analysis (Pontiki et al., 2015), which saw participation from over 25 teams. In Aspect-based Sentiment Analysis, the aim is to identify the aspects of entities and the sentiment expressed for each aspect. In this paper, we present a detailed description of our system, that stood 4th in Aspect Category subtask (slot 1), 7th in Opinion Target Expression subtask (slot 2) and 8th in Sentiment Polarity subtask (slot 3) on the Restaurant datasets.

1 Introduction

When a review or a social media post talks about a product or service, the user might want to discuss multiple aspects or sub-topics related to the product or service being discussed. For example, in a restaurant review, while the customer might have good things to say about the food quality offered at a restaurant, she might be disappointed with the service offered to her, and she might think the decor needs to be revamped. So a general sentiment analyzer that determines the overall sentiment towards the product or service might not be able to capture the full essence of the review. Hence the need for Aspect-based Sentiment Analysis, for better and more fine-grained analysis of user feedback, which would enable service providers and product manufacturers to identify those business aspects that needs improvement. Specifically, SemEval-2015 Task 12 expects systems to automatically determine

the aspect categories present in the data and the sentiment expressed towards each of those categories, given a customer review. For the Aspect Category (Entity and Attribute) Detection subtask, one has to identify every entity E and attribute A pair E#A towards which an opinion is expressed in the given text. E and A should be chosen from predefined inventories of Entity types and Attribute labels per domain. Each E#A pair together defines an aspect category of the given text. The E#A inventories for the restaurants domain has been shown in Table 1.

For the Opinion Target Expression (OTE) identification subtask (Slot 2), we need to identify an expression used in the given text that refers to the reviewed entity E of a pair E#A. The OTE is defined by its starting and ending offsets in the given text. The OTE slot takes the value “NULL” when there is no explicit mention of the opinion entity or no mention at all.

For Sentiment Polarity Detection task, each identified E#A pair of the given text has to be assigned a polarity - positive, negative, or neutral.

2 Related Work

The Aspect Category Detection task can be thought of as similar to document classification task, which has a huge trove of excellent literature. Specifically delving into classification of reviews, (Kiritchenko et al., 2014) showed state-of-art performance, using interesting linguistic and lexicon features. (Castellucci et al., 2014) used simple bag of words based features, generalized using distributional vectors learnt from external data. (Brychcín et al., 2014) employed MaxEnt classifiers using addi-

tional features like word clusters learnt using various methods like LDA.

(Hu and Liu, 2004b) initiated works on aspect identification in product reviews using an association rule based system. In his book (Liu, 2012) specifies four methods for aspect extraction, namely, frequent phrases, opinion and target relations, supervised learning and topic models. (Jakob and Gurevych, 2010) highlighted the use of Conditional Random Fields to extract the aspect terms and phrases and demonstrated a significant improvement in the F-Measure compared to then state-of-the-art by (Zhuang et al., 2006), which used a supervised approach to extract feature-opinion pairs. There are some approaches that utilize NLP semantics to extract aspect terms. Bhattacharyya (Mukherjee and Bhattacharyya, 2012) created a system to discover dependency parsing rules to extract opinion expressions. Many new works use hybrid approaches combining both NLP as well as statistical methods to create improved systems. In SemEval 2014, (Kiritchenko et al., 2014) used an in-house entity tagging system to find labels for Outside Term (O) and Aspect Term (T). (Toh and Wang, 2014) used tagging approach with more linguistic features and extra resources like Wordnet and word clusters.

The task of Sentiment Analysis has been enriched with some of the seminal works like (Pang and Lee, 2004) and (Wilson et al., 2005), and has reached new heights with recent publications from (Socher et al., 2013) which combines grammatical cues with deep learning. (Carrascosa, 2014) presented innovative techniques of ensemble learning for the task of Sentiment Analysis, which we too have adopted in concept. (Bakliwal et al., 2012) presents a simple sentiment scoring function which uses prior information to classify and weight various sentiment bearing words/phrases in tweets. However, none of these works are crafted to handle Aspect based Sentiment Analysis and it is not trivial to adapt them for this task. Similar to the task at hand are works done by (Mcauley et al., 2012) and (Lakkaraju et al., 2011), both of whom mined great benefits from the topic modelling paradigm. (Mohammad et al., 2013) achieved the best performance in Aspect Category Polarity Detection task in SemEval 2014 using various innovative linguistic features and publicly available sentiment lexica and two automatically com-

Entities
RESTAURANT, FOOD, DRINKS, SERVICE, AMBIENCE, LOCATION
Attributes
GENERAL, PRICES, QUALITY, STYLE_OPTIONS, MISCELLANEOUS

Table 1: Entities and Attributes in Restaurants dataset.

puted polarity lexica. (Brun et al., 2014) used information from its syntactic parser, BoW features, and an out-of-domain sentiment lexicon to train an SVM model.

We have experimented with the techniques and features from these previous works and have also added some of our own.

3 Subtask 1: Aspect Category Detection

The Aspect Category Detection task involves identifying every entity E and attribute A pair E#A towards which an opinion is expressed in the given review.

We take a supervised classification approach where we use C one-vs-all Random Forest Classifiers, for each of the C {entity,attribute} pairs or aspect categories in the training data, with basic bag of words based approach. We have also tried other features that we explain shortly, but surprisingly the bag of words approach yielded us the best performance. As a part of the pre-processing procedure, we did the following:

- Removed stop words, except pronouns, because we observed that the category SERVICE#GENERAL can easily be distinguished from other categories by using pronouns as cues
- Stemmed all words
- Removed punctuation
- Normalized all numbers by replacing them by zeros, with the motivation that the exact figures do not hold any semantic meaning and are not of importance to us.

Following is the list of features we experimented with:

- *Unigrams* — For each word in a review, we mark its corresponding position True if it is present in the vocabulary.
- *Presence of number* — We check if a review sample contains numbers or not, with the motivation reviews talking about the PRICES attribute are more likely to have numbers in them.
- *Presence of word in Food and Drinks list*¹ — The motivation behind using this feature is, sentences talking about say, FOOD#PRICES and DRINKS#PRICES are likely to use similar words like “cheap”, “expensive”, “value for money”, “dollars”, etc., but we need to be able to distinguish between the two (FOOD and DRINKS). Hence we use look-up lists for food and drinks with the hope that the customers would explicitly use names of food and drinks items in reviews, wherever applicable.
- *WordNet synsets* — WordNet is a large lexical database of English. In Wordnet, synonyms or words that denote the same concept and are interchangeable in many contexts, are grouped into unordered sets called synsets. Word forms with several distinct meanings are represented in as many distinct synsets, and hence this feature is useful for capturing semantic information. For each word we find its corresponding synset and use it as a feature for our classifier in a bag of words fashion.
- *TF-IDF* — Instead of using binary values to denote absence or presence of a word in the sentence, we put its corresponding TF-IDF score pre-computed from the train data. Normally for document classification tasks, TF-IDF performs better than n-grams because the former rightly penalizes common words that are not helpful in distinguishing one topic from the other. Although our Aspect Detection task is very similar to document classification task, this feature did not help much, probably because of the small size of the data set.
- *Word2Vec* — The Word2Vec is an efficient implementation of skip-gram and continuous

¹Food list compiled from <http://eatingatoz.com/food-list/> and Drinks list compiled manually

bag of words architectures that takes a text corpus as input and produces the word vectors of its constituent words as output. We trained Word2Vec on a corpus comprising Yelp Restaurant reviews data, SemEval 2014 data, SemEval 2015 train data. Let the vector dimension to be D. For each word in a review sentence, we get a vector representation of dimension D. We take an average over all words and end up with a single D-dimensional vector. We experiment with the value of D, which is essentially an optimization over time required to train, and the performance and finally set it to be 30. However, vectors averaged over all words in a sentence are not very good representations for the sentence, which is possibly why this feature did not add much value to our system.

For train and test data were pre-processed and their features extracted in the same way. As for the Random Forest Classifier, we used 50 decision tree estimators using Gini index criterion and at each step we consider only S features when looking for the best split, where S is the square root of the total number of features.

We had also tried hierarchical 2-level classification, i.e. first classifying a review sentence into one of the entities and then classifying them further into one of the pre-defined attributes. However, this 2-level classification technique, with the same set of features mentioned above, yielded poorer performance. So we decided to not make any distinction between entities and attributes, and consider an entity-attribute pair together as an aspect category.

This task required us to categorize reviews into very fine-grained and inter-related categories, with hierarchical dependencies among themselves. This might have been one of the reasons why many of the popular features used for regular document classification did not perform as good as they promised to. Another challenge was the small number of training examples, as compared to the large number of categories to be classified into, which was not the case in any of the previous works to the best of our knowledge.

4 Subtask 2: Opinion Target Expression

Given a review sentence, the aim of this task is to find the Opinion Target Expressions (OTE), that is, the particular attribute of the entity the user expresses his/her sentiment about. Aspects may either be explicitly explained in the review as in the sentence “The service was really quick and I loved the fajitas.” Here “service” and “fajita ”are explicit aspects. In a sentence like “Don’t go. Really horrible”, the user didn’t use any individual term but still gives an impression of her sentiment. In such cases, the slot takes the value “NULL”. Our system uses a sequence labelling approach to tackle this problem by the use of Conditional Random Fields. The tagger from Mallet toolkit, is trained to identify three possible tags, namely BEG and INT for beginning and intermediate target words and OTH for other words.

Our features are as follows:-

- *Word* — The lowercase form of the word itself
- *POS* — Part of speech tag of the word
- *Dependency* — We use two kinds of dependency features — the dependency label on incoming edge on the word, and the first dependency label on outgoing edge. This proved to be a very important feature.
- *Capitalization* — If the first character of the word is in capital, mark it as capital.
- *Punctuation* — If the word contains any non-alphanumeric character, we mark it as punctuation.
- *Seed* — The word is marked as a seed if it was present in the seed-list created by collecting all the OTEs in the training data, splitting them by word and removing all the stop words.
- *Brown Cluster* — Brown Cluster ID is obtained by first training Brown Clustering on the same corpus we described for Word2Vec features in Subtask 1. Brown clustering is a form of hierarchical clustering of words based on the context in which they occur. The intuition behind the method is that a class-based language model where probabilities of words are based

on the clusters of previous words, can overcome the data sparsity problem inherent in language modeling. From brown clustering, for each word in the corpus we get the cluster ID to which it has been assigned. We generate 100 clusters.

- *Presence in Expanded List* — We curated an expanded seed list from the original seed list explained above. We utilized WiBi, which is a taxonomy of Wikipedia pages and categories. We traversed the WiBi Page graph and collected the pages located next to the words (if present) in the seed list. The new list was again split by spaces and punctuation, and stop words were removed. This feature is marked if the term is present in the expanded list.
- *Stop Word* — This feature is marked if the current term is a stop word in English language.
- *Seed Stem* — This feature contains the stemmed form of the original word as obtained from Porter Stemmer.

5 Subtask 3: Sentiment Polarity Classification

In this subtask, the input consists of a review sentence and the set of aspect categories it belongs to. The expected output is a polarity label for each of the associated aspect categories. We have first extracted Bag of Words and Wordnet Synset features from both train and test data. Then we run a variety of classifiers (like Stochastic Gradient Descent, SVM, Adaboost) multiple times and store the confidence scores obtained from decision functions of each of these classifiers. Finally we build a linear SVM classifier that uses the scores obtained from the classifiers in the 1st level as features, along with 15 other hand-crafted lexicon features as explained in Section 5.2. This is also known as stacking, a form of ensemble learning. It is essentially stacking of classifiers inside a classifier. Stacking typically yields performance better than any single one of the trained models, and this is what we wanted to leverage. However, since we need polarity labels per aspect category, we need to identify the segments in the sentence that deals with each of the categories

and then treat those segments as individual samples for polarity detection. For example, if there are three aspect categories associated with a sentence, we want to break it down into 3 {sentence,category} pairs:

$$sent1, \{cat1, cat2, cat3\} \rightarrow \{sent1, cat2\}, \{sent2, cat2\}, \{sent3, cat3\}$$

For each {sentence, category} pair, we find a word in the sentence that is the best representative of the category, which we call as centroid. Then we take a window of n words surrounding the centroid and consider that window to be the segment of interest for that category. So in this example sentence, we need to have three centroids and hence three segments, not necessarily disjoint:

$$\{sent1, cat1\}, \{sent1, cat2\}, \{sent1, cat3\} \rightarrow \{seg1, cat1\}, \{seg2, cat2\}, \{seg3, cat3\}$$

We experimented with the window size, and decided upon using a window size of 3 words to the left and to the right of the centroid. It is interesting to note that among sentences that have more than one category, the average length of a review sentence is 15 words in train data and 17 words in test data.

After we get these segments, we extract the following features from these segments for polarity detection:

- *Bag of Words*
- *Grapheme Stretching* i.e. words with repeated characters. For example, words like “Tooood goood” indicates strong subjectivity and therefore is less likely to belong to Neutral class.
- *Presence of exclamation* also signals subjectivity, usually positivity.
- *Presence of wh-words and conditional words* like why, what, if, etc. Observation tells us that such presence are mostly characteristic of sentences with negative polarity.
- *Wordnet Synsets*, as explained before

While bag of words features include statistical information, WordNet synsets help incorporate semantic information. These two complementary features help us in making the maximal discrimination among the target classes.

5.1 Extracting Centroid for a {Sentence, Category} Pair

We automatically generate a set of seed words for each of the aspect categories by the following technique: From the train data, we consider all sentences labelled with a single category as a single document. As a result, for 13 possible categories in the train data, we have 13 documents. Now for each document (corresponding to each category), we compute the TF-IDF scores of all the words and consider words having TF-IDF greater than a certain threshold as seed words for that category. We ascertain the optimal value of the threshold to be 0.2 through experimentation. We generate a co-occurrence matrix of words from three datasets SemEval 2015 train data, SemEval 2014 train and test data. Typically, it is considered that two words co-occur if they are present as bigram in the corpus. However, we define co-occurrence as occurring in the same review sentence, rather than occurring as a bigram as it is less likely to find repetition of co-occurring bigrams in a smaller corpus. This co-occurrence matrix stores the frequency of co-occurrence of two words in the corpus. For N words in the vocabulary, we have a $N \times N$ co-occurrence matrix. Given a {sentence, category} pair, for each word in the review sentence, we find the Point wise Mutual Information (P.M.I.) between that word and each word in the seed list of the assigned category and take their average for that word. We do the same for all words in the sentence. The word in the sentence having the maximum average P.M.I. score is defined as the centroid for the {sentence, category} pair. P.M.I is defined as the ratio of the probability of occurrence of two words together in the corpus to the product of the probabilities of occurrence of the two words independently in the corpus. We derive the co-occurrence frequencies from the co-occurrence matrix we built in the previous step.

5.2 Ensemble Learning – Stacking Classifiers

After feature extraction, we train 3 kinds of classifiers — Linear Support Vector Machines, Stochastic Gradient Descent and Adaboost, for each of the features — Bag of words and Wordnet Synsets. We repeat the process K times where $K \in \mathbb{Z}$. We have experimentally chosen K to be 30 — it is ac-

tually a trade off between the time taken to train the model and the performance improvements. As we increased K over 30, the improvement in performance started to diminish. For each test sample, we obtain 3 scores (corresponding to three classes — positive, negative, neutral) from the decision function of each classifier. We use these confidence scores as features along with 15 other hand-crafted lexicon features for a linear Support Vector Machine classifier. We employ features such as number of positive tokens, number of negative tokens, total positive sentiment score, total negative sentiment score, sum of sentiment scores, maximum sentiment score, etc. from Sentiwordnet (Baccianella et al., 2010), Bing Liu’s opinion lexicon (Hu and Liu, 2004a), MPQA subjectivity lexicon (Wilson et al., 2005), NRC Emotion Association lexicon (Mohammad and Turney, 2013), Sentiment140 lexicon (Go et al., 2009), and NRC Hashtag Lexicon (Mohammad and Kiritchenko, 2014).

For the final linear SVM classifier, we experimentally ascertain the optimal value of the parameter C to be 0.024. The linear SVM classifiers, in the first level of stacking, had a default value of 1.0 for parameter C. We did not have enough time to tune them, as we had many classifiers inside the main SVM classifier. The Ada Boost Classifier uses 100 decision tree estimators and a default learning rate of 1. We have used Scikit Learn for building all the classifiers. Although we employ several classifiers, the time taken is negligible. This is because the different classifiers in the first stage of stacking can be trained in parallel quite easily.

6 Results

We submitted unconstrained systems for the Restaurants dataset. We did not run our system for other domains mainly due to lack of time during the competition. Table 2 shows our final F1-scores obtained on SemEval official test data, for each of the three slots. Tables 3, 4 and 5 presents the results of ablation experiments carried out for slots 1, 2 and 3 respectively. We show the effect of varying the size of the context window surrounding the centroid, on F1-score in Figure 1. Finally Table 6 compares our ensemble system with a baseline system trained on a single linear SVM with only lexicon features.

Subtask	Our Score	Best Score	Rank
Slot 1	0.57	0.62	4
Slot 2	0.53	0.70	7
Slot 3	0.71	0.78	8

Table 2: Official Results for SemEval 2015.

Feature	Precision	Recall	F1
Unigrams	0.64	0.51	0.57
Unigrams+Bigrams	0.51	0.45	0.48
Unigrams+WordNet syn	0.53	0.48	0.50
Unigrams+Word2Vec	0.52	0.46	0.49
TF-IDF	0.47	0.42	0.44

Table 3: Experiment with Features for Slot 1.

Feature	Precision	Recall	F1
All	0.51	0.55	0.52
All - (Seed+Expanded Seed)	0.52	0.55	0.53 ²
POS+Dep.+Punct.+Brown	0.35	0.54	0.43
POS+Dep.+Punct.+Stopwords	0.64	0.53	0.58 ³
POS+Dep.	0.62	0.51	0.57

Table 4: Ablation Experiment for Slot 2.

Feature	Accuracy
All (BoW + WordNet syn)	0.71
All - BoW	0.68
All - WordNet syn	0.70

Table 5: Ablation Experiment for Slot 3.

System	Accuracy
Linear SVM with only lexicon	0.68
Our system	0.71

Table 6: Slot 3: Comparison of our ensemble learning technique with a baseline system trained on a single LinearSVM with only lexicon features.

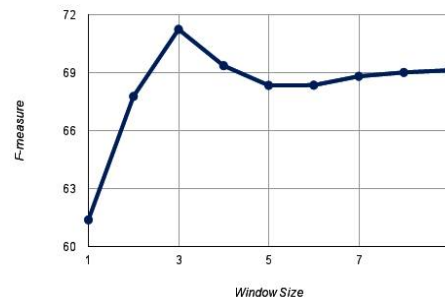


Figure 1: Variation of F1 score with context window size.

²Submitted system

³This result was obtained during ablation experiment post-competition

7 Conclusion

This paper describes the system submitted by team SIEL for SemEval 2015 Task 12. For all the three subtasks, our system performs quite well, ranking between 4th and 8th. We experimented with Ensemble Learning technique for slot 3, which we want to explore and improve further. In future, we would like to work on adapting our system to other domains as well.

Acknowledgements

We sincerely thank Samik Datta and Mohit Kumar for all the fruitful discussions and encouragement, Riddhiman Dasgupta for insights on Ensemble Learning and Mark Franco-Salvador for guiding us with Wibi.

References

- Stefano Baccianella, Andrea Esuli, and Fabrizio Sebastiani. 2010. Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In *Proc. of LREC*.
- Akshat Bakliwal, Piyush Arora, Senthil Madhappan, Nikhil Kapre, Mukesh Singh, and Vasudeva Varma. 2012. Mining sentiments from tweets. In *Proceedings of the 3rd Workshop in Computational Approaches to Subjectivity and Sentiment Analysis*, WASSA '12, pages 11–18, Stroudsburg, PA, USA.
- Caroline Brun, Nicoleta Diana Popa, and Claude Roux. 2014. Xrce: Hybrid classification for aspect-based sentiment analysis. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 838–842.
- Tomáš Brychcín, Michal Konkol, and Josef Steinberger. 2014. Uwb: Machine learning approach to aspect-based sentiment analysis. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 817–822.
- Rafael Carrascosa. 2014. An entry to kaggle’s ‘sentiment analysis on movie reviews’ competition.
- Giuseppe Castellucci, Simone Filice, Danilo Croce, and Roberto Basili. 2014. Unitor: Aspect based sentiment analysis with structured learning. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 761–767, Dublin, Ireland, August.
- Ingo Feinerer and Kurt Hornik, 2014. *wordnet: WordNet Interface*. R package version 0.1-10.
- Jenny Rose Finkel, Trond Grenager, and Christopher Manning. 2005. Incorporating non-local information into information extraction systems by gibbs sampling. In *In ACL*, pages 363–370.
- Alec Go, Richa Bhayani, and Lei Huang. 2009. Twitter sentiment classification using distant supervision. *Processing*, pages 1–6.
- Minqing Hu and Bing Liu. 2004a. Mining and summarizing customer reviews. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '04, pages 168–177, New York, NY, USA.
- Minqing Hu and Bing Liu. 2004b. Mining opinion features in customer reviews. In *Proceedings of the 19th National Conference on Artificial Intelligence*, AAAI'04, pages 755–760.
- Niklas Jakob and Iryna Gurevych. 2010. Extracting opinion targets in a single- and cross-domain setting with conditional random fields. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, EMNLP '10, pages 1035–1045, Stroudsburg, PA, USA.
- 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.
- Himabindu Lakkaraju, Chiranjib Bhattacharyya, Indrajit Bhattacharya, and Srujana Merugu, 2011. *Exploiting Coherence for the Simultaneous Discovery of Latent Facets and associated Sentiments*, chapter 43, pages 498–509.
- Bing Liu. 2012. *Sentiment Analysis and Opinion Mining*. Synthesis Lectures on Human Language Technologies.
- Bing Liu. May 2012. *Sentiment Analysis and Opinion Mining (Introduction and Survey)*.
- Edward Loper and Steven Bird. 2002. Nltk: The natural language toolkit. In *In Proceedings of the ACL Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics*. Philadelphia: Association for Computational Linguistics.
- Julian McAuley, Jure Leskovec, and Dan Jurafsky. 2012. Pale lagerthe pale lager model.
- Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781.
- Saif M. Mohammad and Svetlana Kiritchenko. 2014. Using hashtags to capture fine emotion categories from tweets. *Computational Intelligence*, pages n/a–n/a.
- Saif M. Mohammad and Peter D. Turney. 2013. Crowdsourcing a word-emotion association lexicon. 29(3):436–465.

- Saif M. Mohammad, Svetlana Kiritchenko, and Xiaodan Zhu. 2013. Nrc-canada: Building the state-of-the-art in sentiment analysis of tweets. *CoRR*, abs/1308.6242.
- Subhabrata Mukherjee and Pushpak Bhattacharyya. 2012. Feature specific sentiment analysis for product reviews. In Alexander Gelbukh, editor, *Computational Linguistics and Intelligent Text Processing*, volume 7181 of *Lecture Notes in Computer Science*, pages 475–487.
- Bo Pang and Lillian Lee. 2004. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the 42Nd Annual Meeting on Association for Computational Linguistics*, ACL '04, Stroudsburg, PA, USA.
- 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, Dimitris Galanis, John Pavlopoulos, Harris Papageorgiou, Ion Androutsopoulos, and Suresh Manandhar. 2014. Semeval-2014 task 4: Aspect based sentiment analysis. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 27–35, Dublin, Ireland, August.
- Maria Pontiki, Dimitris Galanis, John Pavlopoulos, Harris Papageorgiou, Suresh Manandhar, and Ion Androutsopoulos. 2015. Semeval-2015 task 12: Aspect based sentiment analysis. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, Denver, Colorado.
- Radim Řehůřek and Petr Sojka. 2010. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta, May.
- Sara Rosenthal, Alan Ritter, Preslav Nakov, and Veselin Stoyanov. 2014. Semeval-2014 task 9: Sentiment analysis in twitter. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 73–80, Dublin, Ireland, August.
- Richard Socher, Alex Perelygin, Jean Y. Wu, Jason Chuang, Christopher D. Manning, Andrew Y. Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank.
- Zhiqiang Toh and Wenting Wang. 2014. Dlirec: Aspect term extraction and term polarity classification system. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 235–240, Dublin, Ireland, August.
- Joachim Wagner, Piyush Arora, Santiago Cortes, Utsab Barman, Dasha Bogdanova, Jennifer Foster, and Lamia Tounsi. 2014. Dcu: Aspect-based polarity classification for semeval task 4. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 223–229, Dublin, Ireland, August.
- Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. 2005. Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, HLT '05, pages 347–354, Stroudsburg, PA, USA.
- Li Zhuang, Feng Jing, and Xiao-Yan Zhu. 2006. Movie review mining and summarization. In *Proceedings of the 15th ACM International Conference on Information and Knowledge Management*, CIKM '06, pages 43–50, New York, NY, USA.