Targeted Sentiment to Understand Student Comments

Charles Welch and **Rada Mihalcea**
University of Michigan
2260 Hayward Street
Ann Arbor, MI 48109, USA
{cfwelch,mihalcea}@umich.edu

Abstract

We address the task of targeted sentiment as a means of understanding the sentiment that students hold toward courses and instructors, as expressed by students in their comments. We introduce a new dataset consisting of student comments annotated for targeted sentiment and describe a system that can both identify the courses and instructors mentioned in student comments, as well as label the students' sentiment toward those entities. Through several comparative evaluations, we show that our system outperforms previous work on a similar task.

1 Introduction

Sentiment analysis is the computational study of people's opinions or emotions; it is a challenging problem that is increasingly being used for decision making by individuals and organizations (Pang and Lee, 2008). There is a significant body of research on sentiment analysis, addressing entire documents (Agarwal and Bhattacharyya, 2005), including blogs (Godbole et al., 2007; Annett and Kondrak, 2008) and reviews (Yi et al., 2003; Cabral and Hortacsu, 2010); sentences (Yu and Hatzivassiloglou, 2003; Nigam and Hurst, 2004) or otherwise short spans of texts such as tweets (Pak and Paroubek, 2010; Kouloumpis et al., 2011); and phrases (Wilson et al., 2005; Turney, 2002). More recent work has also addressed the task of aspect sentiment (Pontiki et al., 2015; Thet et al., 2010; Lakkaraju et al., 2014), which aims to address the sentiment toward attributes of the target entity, such as the service in a restaurant (Sauper and Barzilay, 2013), or the camera of a mobile phone (Chamlertwat et al., 2012).

In this paper we address the task of *targeted sentiment*, defined as the task of identifying the sentiment (*positive, negative*) or lack thereof (*neutral*) that a writer holds *toward* entities mentioned in a statement. Targeted sentiment has been only recently introduced as a task, to our knowledge with contributions from only two research groups that focused primarily on settings with scarce resources (Mitchell et al., 2013; Zhang et al., 2015). While previous work on data sets such as product reviews can give an accurate measure of sentiment toward products (as explicit targets of the opinions being expressed in the reviews), some corpora include additional challenges. Targeted sentiment addresses the challenge of identifying entities in running text (e.g., Twitter, student comments), and attributing separate sentiment to each mentioned entity.

In our work, we focus on an application-driven task, namely that of understanding students' sentiment towards courses and instructors as expressed in their comments. As an example, consider the statement:

(1) *I thought that natural language processing with professor Doe was a great class.*

We want to recognize the targets "natural language processing" (a course) and "Doe" (an instructor), as well as a positive sentiment toward the course, and a neutral sentiment toward the instructor. We approach targeted sentiment as a pipeline of two tasks: (1) entity extraction, which aims to identify the entities of interest (in our case, courses and instructors); and (2) entity-centered sentiment analysis, which classifies the sentiment (positive, negative, neutral) held by the student writer toward those entities.

Section 2 overviews previous work, and shows how our work fits into the bigger picture of sentiment analysis research. Section 3 describes the data used for our experiments. Section 4.1 shows how entities are extracted from text for use in targeted sentiment analysis, and Section 4.2 describes how the sentiment held toward these entities is classified. An overall evaluation of our system and comparison with previous work are presented in Section 5, followed by a discussion and conclusions in Section 6.

2 Previous Work

Most work in sentiment analysis is done at one of three levels: document level, sentence level, and aspect level. These three levels of granularity are ordered from coarsest to finest, with the finer granularity tasks being less well studied. In general, an opinion can be represented by the following quintuple, $(e_i, a_{ij}, oo_{ijkl}, h_k, t_l)$ (Zhang and Liu, 2014). The value e_i here represents the ith entity and a_{ij} represents the aspect j of this entity. The kth holder of the opinion is represented by h_k and the time, l, that the opinion is expressed is given by t_l. Given the entity, aspect, holder, and time, one can reason about an opinion orientation oo_{ijkl}. This is usually a positive, negative, or neutral value, although occasionally a larger number of sentiment values are used (e.g., very positive, very negative).

The work most similar to ours is the open domain targeted sentiment task (Mitchell et al., 2013). Unlike Mitchell et al, we do not use an artificially balanced data set. Instead we collected all the utterances from students who talked about whichever entities they chose. While we do limit the types of entities to only classes or instructors, we do not limit the specific entities themselves and students can talk about any entities that are relevant to their previous educational experience. Our method is also somewhat different in that we do not evaluate subjectivity: all the entities are assigned a positive, negative, or neutral sentiment, and there are no entities without sentiment.

There were two follow-up papers to Mitchell et al. (Zhang et al., 2015; Zhang et al., 2016) (both from the same research group). The first of these papers worked on improving the three models used in Mitchell et al. including the pipeline, joint, and collapsed models. They show some improvements but the pipeline mode, which is most similar to ours, does not greatly differ in performance. The latter paper used different neural network models on a combination of three data sets. Two of these data sets are derived from Twitter (including Mitchell's) and the last is derived from MPQA. We do not attempt to compare to that work, but we show comparable F1 measures using simple linguistic features.

The next most closely related work to ours are the tasks of sentiment slot filling, target dependent sentiment analysis, and aspect-based sentiment analysis. Slot filling is the task of discovering information about a named entity and storing it in a knowledge source (Surdeanu and Ji, 2014). The 2013 Text Analysis Conference (TAC) had two similar tasks, which were slot filling and temporal slot filling (Surdeanu, 2013). For the slot filling task, systems had to determine the correct value for a set of slots for people and organizations. People contained slots such as "date of birth", "age", or "spouse", while organizations contained slots such as "website", "founded by", and "country of headquarters". For the task of temporal slot filling, a system must determine two time ranges. The first time range is a range in which the expressed slot-value was known to begin being a true statement and the second time is when the expressed fact is known to have ceased being true. Sentiment slot filling is the task of taking a query opinion holder and orientation and returning the set of entities that satisfy this condition. In terms of the quintuple we use to represent sentiment, these are related tasks because although slot filling and temporal slot filling are not exactly sentiment tasks, they are concerned with the entity, aspect, and time values. The sentiment slot filling task is concerned with the entity, orientation, and opinion holder.

In the task of target dependent sentiment analysis, the goal is to take a query entity and find the sentiment toward this entity in a set of tweets (Jiang et al., 2011). This is usually done with a small set of entities where the corpus is constructed by querying Twitter for tweets that contain the substring matching the entity. Jiang et al. perform this task in three steps. They first identify whether subjectivity exists, then the polarity of the sentiment toward the target, and then use a graph based method to improve classification accuracy using retweets, i.e., tweets from the same users that mention the same entities. In our task, we use a larger set of entities that have many ways of being mentioned; this makes the entity identification part of the task more difficult. We also do not have a social network structure to leverage to improve performance.

Aspect-based sentiment analysis has been the focus of recent SemEval tasks as well as a TAC task (Ellis et al., 2014; Pontiki et al., 2014; Pontiki et al., 2015). The 2014 sentiment task was continued in 2015, and again in 2016. Researchers submitted a variety of models to evaluate the sentiment of aspects on sets of reviews for laptops, restaurants, and hotels. The highest scoring systems in the SemEval 2015 Task 12 used maximum entropy and support vector machine (SVM) models with bag of words (BoW), verb and adjective lemmas, bigrams after verbs, negation terms, punctuation, point-wise mutual information scores, part of speech tags, and other features (Li et al., 2013; Zhang and Lan, 2015). The results presented were marked as either constrained or unconstrained systems. Unconstrained systems were allowed to use data outside the training data provided, while constrained systems could not. The top two scoring models were unconstrained but the top scoring constrained system used Brown clusters in addition to other features. These are counts of how many words in the sentence belong to semantic clusters of words derived in previous work (Hamdan et al., 2015). Other entries used similar features with several entries using SVM models and a single entry that relied on an unsupervised model.

3 Data

As we are not aware of any dataset consisting of statements describing courses and instructors, and the sentiment that the writers (students) have toward them, we collected our own dataset. We extracted sentences from a Facebook student group where students describe their experience with classes in the Computer Science department at the University of Michigan, as well as from a survey run with students in the same department. The final data set consists of 1,042 utterances written by both graduates and undergraduates, describing both classes and instructors that the students had/interacted with. Table 1 shows three statement examples drawn from our dataset.

Student utterance	Annotation
I thought that introductory programming concepts was a difficult class and I did not like it.	⟨class name=introductory programming concepts, sentiment=negative⟩
Professor Williams is my favorite teacher that I've had so far.	⟨instructor name=Williams, sentiment=positive⟩
I took CS 203 last Winter. Davis was teaching and I thought the class was excellent.	⟨class name = CS 203, sentiment=positive⟩ ⟨instructor name=Davis, sentiment=neutral⟩

Table 1: Sample student utterances from our dataset along with annotations.

All the utterances were first manually annotated by one of the authors to identify courses and instructors. As often done in entity extraction methods, we identify entities using an I(nside) O(utside) B(eginning) model. For instance, given the text "I am enrolled in CS 445.", and assuming the entity to be extracted is a course name, the annotation would include the following labels "I_O am_O $\mathrm{enrolled}_O$ in_O CS_B 445_I.", indicating that CS is at the beginning of the course name, 445 is inside a course name, and all the other tokens are outside the course name.

Classes can be mentioned by department and class ID as in "CS 484," by ID alone as in "484," or by name as in "introduction to artificial intelligence" or "intro to AI." Instructors are mentioned by name, but could be mentioned by first, last, or first and last names. In total, the 1,042 utterances include 976 class mentions and 256 instructor mentions, for a total of 1,232 entities.

The perceived sentiment toward each entity was also manually labeled by one of the authors as either positive, negative, or neutral. When no explicit sentiment is expressed toward an entity, it is assumed to be neutral. If no sentiment is evident from a given utterance, it is assumed to be neutral. Table 1 shows the annotations for the three sample utterances from our dataset.

To calculate inter-annotator agreement for the identification of entities, a second annotator labeled 100 utterances from the data set, containing 1,263 tokens. Of these, 1,067 were mutually labeled as not being part of any entity. Of the remaining 196 tokens, 2% were not in agreement. Including all tokens, agreement was measured as 0.987 using Cohen's kappa. These two percent were two instances where the human judges disagreed on whether or not a sequence of tokens was a course name (i.e., an entity that needed to be annotated) or simply a course description. For example, in the sentence "I believe that databases are a crucial part of computer science and 520 was interesting," while "databases" is part of the class name, one annotator decided that the word was simply a description of the content of the course and not an entity.

To calculate inter-annotator agreement for sentiment annotations, a second annotator individually labeled all the 1,232 entities. The agreement between the two annotators was measured at 77.7%, which gives a Cohen's kappa of 0.661 considered to be good agreement. Agreement was calculated as the percentage of entities for which both annotators assigned the same label. Of the annotator disagreements, 10.7% were neutral-negative disagreements, 11.2% neutral-positive disagreements, and 0.2% positive-negative.

4 Targeted Sentiment Analysis

We address this task as a pipeline of two steps. We first identify the target entities (i.e., courses and instructors), followed by a classification held by the student writer toward those entities. In the following, we describe and evaluate the method used for each step, and compare the results obtained against the state-of-the-art.

4.1 Entity Extraction

As mentioned before, we use an IOB model to identify entities in the text. We therefore apply a classification process to every token in the input text. For each token, we build a feature vector, using the following features:

Core features. These include the current word, the case and part-of-speech of the current word, the previous two words; features are also derived from the two words neighboring the current word, which are computed the same way as for the current word.

Lexicons. We record the presence/absence of words in two custom lexicons: one consisting of the professor names gathered from the University of Michigan; the second one including all the words used in the names of the classes offered in the Computer Science department at the same university. The lexicon features are generated for the current word as well as each neighboring word.

Professor titles. We use a list of titles, such as "Dr." or "Prof." to assist with the identification of professor names. The list was compiled manually, and consists of 15 tokens. A feature is generated to indicate whether a token belongs to this list or not. 38% of utterances in the corpus contain professor titles.

Sequence. Students often use a subset of the words in a class name to refer to it. The sequence feature is a binary feature that indicates whether the current word is inside a course or an instructor name sequence, where the courses and instructor names are drawn from the two lexicons described above.

Acronym. The acronym feature is another binary feature that indicates if the input token is an acronym of any class or instructor names in the lexicons. It takes the first letters of each of the words in a name and checks to see if the token matches the concatenated string of first letters for an entry in the lexicon. It subsequently checks if the removal of any number of letters, while retaining order, matches the given token. For instance, "AI" and "ITAI" both match "Introduction to Artificial Intelligence".

Nearest entity. Sometimes class or instructor names are misspelled, and for such cases lexicon features may not be effective. We create a feature that checks if the current token has an edit distance less than three to a word in a class or instructor name in the lexicons. If a match is found, the feature is set to a value of "C" (course) or "I" (instructor) respectively. If no token exists, the feature is set to "N".

As a machine learning algorithm, we use a conditional random field, as it has been previously shown to be highly effective for such entity extraction tasks (Zhang and Liu, 2014). We run a set of 67-33 train-test splits using stratified sampling. Table 2 shows the F-measure results obtained by our system, which makes use of all the features described above, for each of the four token types (B and I for courses and instructors). For comparison, we also show the results obtained with a basic setting, when only the core features are used, as well as the results obtained with a state-of-the-art entity extraction system available from the Stanford NLP group (Finkel et al., 2005), which we have retrained using our corpus.

Using our system, we see a statistically significant improvement over the core baseline for all four tokens ($p < 0.01$). We also find a statistically significant improvement over the Stanford system for I_C and B_I ($p < 0.01$) but no significant difference for the other two token types.[1]

System	B_C	I_C	B_I	I_I
Our system	0.945	0.881	0.922	0.901
Baseline (core features)	0.940*	0.849*	0.863*	0.841*
Stanford (Finkel et al., 2005)	0.944	0.848*	0.896*	0.908

Table 2: F-score figures for the identification of I and B tokens, for course (C) and instructors (I), where * indicates a that our system has a statistically significant improvement for the given token ($p < 0.01$)

To gain a better understanding of the role played by each of the features considered, we also perform feature ablation, with results for the individual feature sets shown in Table 3. We also show the base feature set for comparison.

Interestingly, while lexicon features show the greatest improvement, the titles feature does not show any improvement over the base features. It is possible that this feature ends up being subsumed by the neighboring words, included in the base features. The sequence, acronym, and nearest entity features are all based on the provided lexicons so it is not surprising that sequence and nearest entity features work well. Among them, the acronym feature appears to be less useful simply because many class names are not commonly abbreviated. The most frequently abbreviated name is “AI” for “artificial intelligence”. Classes are more often referred to by a subset of the words in the class name, which is a case covered by the sequence feature. This is why we see an improvement in I tokens for classes, whereas the instructor I tokens do not show an improvement for these features. There are also fewer I instructor tokens overall in the corpus, which could make it harder to learn the importance of these features.

Since classes can be identified by an ID number (e.g., “490”) or by a name (e.g. “Machine Learning”) we can examine the B_C token in more detail. If we separate the B_C token into a token for class IDs and a token for class name words, we find that the improvement using the lexicon, sequence, and nearest entity features is statistically significant only for the class name words ($p < 0.01$). There is no statistically significant improvement for the ID tokens by themselves, which is not surprising given that the identification of such IDs (most of the times consisting of numbers) is an easy task.

Features	B_C	I_C	B_I	I_I
Baseline (core features)	0.940	0.849	0.863	0.841
Lexicons	0.944*	0.875*	0.915*	0.896*
Titles	0.940	0.851	0.861	0.839
Sequence	0.945*	0.871*	0.858	0.832
Acronym	0.940	0.848	0.860	0.835
Nearest entity	0.944*	0.865*	0.910*	0.895*

Table 3: Feature ablation for the identification of I and B tokens, for courses (C) and instructors (I). A feature that provides results significantly better than the base feature set is indicated with * ($p < 0.01$)

We also run an entity-based evaluation, where we use the IOB tokens to construct full class and instructor names. This is done by finding the B tokens that have the correct following sequence of I tokens. If any of the B or I tokens are missing, or are of the wrong type, the entity is not counted as correct. Table 4 shows the precision, recall, and F-score obtained by our system for the extraction of instructor and class entities, and compares our results with those obtained with the Stanford entity extraction system.

[1] Throughout this paper, we measure the statistical significance of our results by using a paired t-test with Bonferroni correction using the same 67-33 train-test splits.

Set	Our system			Stanford (Finkel et al., 2005)		
	Precision	Recall	F-score	Precision	Recall	F-score
Instructors	0.833	0.888	0.859	0.711	0.802	0.754
Classes	0.920	0.899	0.910	0.886	0.867	0.876
Both	0.900	0.897	0.900	0.845	0.853	0.849

Table 4: Precision, Recall and F-score measures for the identification of class and instructor entities

4.2 Entity-Centric Sentiment Analysis

Once the entities of interest are identified, the next step is to determine the sentiment held by the writer (student) toward those entities. This is performed as a classification task using three classes: positive, negative, and neutral. For each candidate entity, we build a feature vector using one of the following configurations:

Weighted bag-of-word. The default model is constructed using unigram counts. The first step is to extract a set of the words that exist in the training set. Using this vocabulary set, counts are constructed for every utterance. These counts are weighted based on their distance, in number of tokens, to the target entity in the statement. For each occurrence of each word, the feature is computed by $\sum_{i \in I} 1/d_{ie}$, where I is the set of occurrences of that word and d is the distance (in words) to the target entity e.

Tree weighted n-grams. A sentence is not linear in nature. A sentence contains clauses and phrases that can be grouped into a tree structure. Consider the sentence "I thought that CS 203 was going to be good, but it was awful". In this sentence "CS 203" is the target entity and we find that a positive sentiment word "good" is closer (using linear distance in number of tokens) to the entity than the negative sentiment word "awful," which represents the actual sentiment toward the entity. If we construct a constituency parse tree from this sentence, and calculate the distance as the number of hops between nodes in the tree, then the negative sentiment word is actually closer to the entity word. For each word in an utterance, we calculate this feature as the number of edges in the parse tree between that word and the target entity. For instance, for the example shown in Figure 1, the distance between "awful" and the target entity "203" is six, while the distance between "good" and "203" is eight.

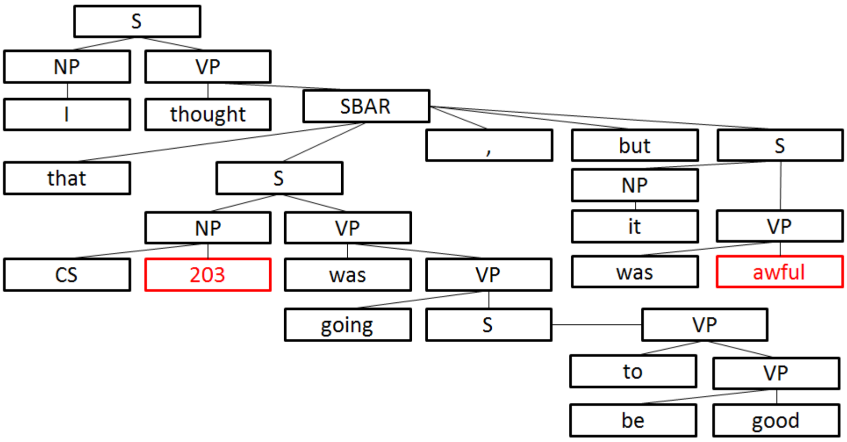

Figure 1: Example sentence, "I thought that CS 203 was going to be good, but it was awful", showing the parse tree weighting for counts using the number of node hops between a given word and the target entity.

Weighted sentiment lexicons. We also implement a feature based on the presence/absence of words from two sentiment lexicons: Bing Liu's lexicon (Hu and Liu, 2004), and the MPQA lexicon (Riloff and Wiebe, 2003) (Wilson et al., 2005). These are two of the most commonly used lexicons in recent sentiment work, and contain 6,789 and 8,222 words respectively, labeled as positive, negative, or neutral. For each word in the utterance, we now generate four features: one simply reflecting the weight of the word (calculated as described before, as a distance to the target entity), and the other three reflecting whether the word appears as a positive, negative, or neutral word in any of the lexicons; these three latter features are again represented as weighted distance scores.

We use an SVM classifier, with a grid search for the SVM cost and gamma parameters performed using three-fold cross validation on the training set. Training and test splits contained approximately 67% and 33% of the data respectively, stratified as mentioned in Section 4.1, such that the two entity types (instructor or class) and each of the three sentiment class labels are roughly evenly spread across the train, development, and test sets.

Table 5 shows the results obtained with our sentiment analysis system. Note that all the experiments are run on the gold standard set of entities (i.e., manually annotated entities). For comparison, we also report a majority baseline, calculated as the percentage of instances in the entire data set that are neutral, as well as the inter-annotator agreement, as described in Section 3.

We also include the result obtained by using the Stanford sentiment analysis tool (Socher et al., 2013). We do not retrain this model on our own data, as this would require additional node level annotation for the parse tree of each utterance; instead, we use their sentence level sentiment analysis, which assigns an integer score of 0-4 to each sentence, ranging from "very negative" to "very positive". We assign the sentence level scores to each entity contained within that sentences. The five values can be mapped to the three values used in our data set in a number of ways, but the way that maximizes the accuracy over our entire data set maps 0 to our "negative," 1 and 2 to our "neutral," and 3 and 4 to our "positive."

Feature	Accuracy
Our system	69.5%
Majority baseline	52.8%
Stanford (Socher et al., 2013)	62.3%
Annotator agreement	77.7%

Table 5: Sentiment accuracies for our system compared to a majority baseline, the Stanford sentiment analysis tool using recursive neural tensor networks, and the inter-annotator agreement.

For a deeper analysis, Table 6 shows the results obtained by our various features.

Feature	Accuracy
Weighted bag-of-words	67.9%
Tree weighted n-grams	65.6%
Weighted sentiment lexicon	69.5%*

Table 6: Sentiment accuracies of different feature models where * indicates a feature whose difference from the default linear weighted bag-of-words is a statistically significant improvement ($p < 0.01$).

5 Overall Evaluation and Discussion

In the previous section, we described the methods used for each of the two stages of targeted sentiment analysis, along with results obtained at each stage. We now perform an overall evaluation of this task, and compare our system with previous methods for targeted sentiment analysis.

First, we evaluate the correctness of the sentiment at entity level, where an entity is marked as correct only if both the entity and the writer's sentiment toward that entity are correct. Table 7 shows the precision, recall, and F-score obtained for instructor and classes individually, and for all the entities together, assuming: (1) ground truth identification of the entities (i.e., manual annotations); and (2) automatic annotation of the entities using our system from Section 4.1.

Second, we compare the results of our system with previous work by (Mitchell et al., 2013). In that work, the authors use a dataset consisting of 2,350 English tweets containing 3,577 volitional entities, which include PERSON and ORGANIZATION entities. They evaluate the performance of the sentiment on entities by checking only the "B" token from the IOB annotation to see if the associated sentiment is

Entities	Precision	Recall	F-score
Ground Truth Instructors	0.643	0.643	0.643
Ground Truth Classes	0.710	0.710	0.710
Ground Truth Both	0.695	0.695	0.695
Extracted Instructors	0.581	0.578	0.580
Extracted Class	0.571	0.599	0.585
Extracted Both	0.573	0.600	0.586

Table 7: Micro-averaged Precision, Recall, and F-score for full targeted sentiment analysis, for both courses and instructors, using ground truth or automatically identified entities.

correct. If so, it is counted as a true positive. Note that this is less constrained than our evaluation, which also requires that the subsequent I tokens be correct.

In order to allow for a comparison between our system and theirs, we train our pipeline model on their data, by using the same ten-fold cross validation that the previous authors provided. Note that for this comparison, in the entity extraction step of our system we do not use the lexicon, professor title, acronym, sequence, or nearest entity features because of their domain specificity (these features are specifically aimed at finding sentiment toward courses and instructors, and are not expected to be useful on a dataset of general Twitter data). The results of this comparison are shown in Table 8. Mitchell et al. examine targeted sentiment with only volitional entities and do not use “neutral” as a class for targeted sentiment. For these reasons we include the second and third rows in Table 8.

Additionally, because previous work had purposefully not used certain features so that their method could be applied to low resource languages, we also show the performance of the system when we remove the part-of-speech features from our entity extraction step. Note that some of the previous work used accuracy, while other work used F-score; we therefore report both.

We also compare our system to that of (Zhang et al., 2015). Zhang et al. 2015 use a neural network model and report their F-score performance on the same corpus. They perform two evaluations, one that uses only positive/negative sentiment, and one that includes the neutral class. We find that our model is comparable when part-of-speech tags are excluded, but outperform the neural models when they are included.

Method	Accuracy	F-score
Our system	68.3%	0.687
Our system, positive/negative sentiment only	68.6%	0.664
Our system, volitional entities, positive/negative sentiment only	70.8%	0.703
Our system, no part-of-speech features	28.9%	0.393
(Mitchell et al., 2013)	30.8%	NA
(Zhang et al., 2015)	NA	0.401
(Zhang et al., 2015) positive/negative sentiment only	NA	0.279

Table 8: Accuracy and F-score for different versions of our system, as compared to previous work.

Discussion. There are a number of errors that are made by our system. Some of the errors come simply from fully or partially missing entities in the beginning of the pipeline. For instance, we found that the named entity recognition fails on some professor names, mainly because some professors use names other than those listed in the online resources that we used to generate our lexicons. A few other less common errors included recognizing first and last names as separate people, and combining class names listed after each other, e.g. “natural language processing and compilers.”

Another batch of errors have correctly recognized entities, but incorrectly classified sentiment. The most common of these cases is incorrectly assigning the neutral class to an entity; the classifier may be somewhat bias toward this class given that it is assigned to 52% of entities in the corpus. Another error involves having multiple entities in a sentence and assigning the sentiment expressed to the wrong entity. For example, in the sentence “I think that John Smith was an interesting teacher in natural language processing”, positive sentiment is incorrectly assigned to “natural language processing.” Another type of error comes from unresolved pronouns. In the utterance, “John Smith taught the data mining class that I took. He was an amazing teacher and I wish that he would teach machine learning,” “John Smith” is classified as having neutral sentiment, rather than positive; coreference resolution could help if we reweighted the features taking into account the correct pronoun set as entity words.

6 Conclusions

In this paper, we addressed the task of targeted sentiment analysis in the context of understanding the sentiment that students hold toward courses and instructors. We introduced a new annotated dataset, collected from students at the University of Michigan, and proposed new features for the extraction of entities and the classification of the sentiment toward these entities. We performed evaluations of each of the two stages in our pipeline model, and showed that both our entity extraction method and the entity-centric sentiment analysis have performance that is competitive with the state-of-the-art. Moreover, in an overall evaluation of our pipeline, we showed that our system exceeds the performance of the two previously proposed systems for targeted sentiment analysis (Mitchell et al., 2013; Zhang et al., 2015).

Through several feature ablation analyses, we found that lexicon features play an important role in this task, and we plan to further investigate the use of such lexicons in the future, as well as that of more advanced representations of domain-specific knowledge such as knowledge-graphs.

Acknowledgements

This material is based in part upon work supported by IBM under contract 4915012629 and by the Michigan Institute for Data Science. Any opinions, findings, conclusions or recommendations expressed above are those of the authors and do not necessarily reflect the views of IBM or the Michigan Institute for Data Science.

References

Alekh Agarwal and Pushpak Bhattacharyya. 2005. Sentiment analysis: A new approach for effective use of linguistic knowledge and exploiting similarities in a set of documents to be classified. In *Proceedings of the International Conference on Natural Language Processing (ICON)*.

Michelle Annett and Grzegorz Kondrak. 2008. A comparison of sentiment analysis techniques: Polarizing movie blogs. In *Conference of the Canadian Society for Computational Studies of Intelligence*, pages 25–35. Springer.

Luis Cabral and Ali Hortacsu. 2010. The dynamics of seller reputation: Evidence from ebay. *The Journal of Industrial Economics*, 58(1):54–78.

Wilas Chamlertwat, Pattarasinee Bhattarakosol, Tippakorn Rungkasiri, and Choochart Haruechaiyasak. 2012. Discovering consumer insight from twitter via sentiment analysis. *J. UCS*, 18(8):973–992.

Joe Ellis, Jeremy Getman, and Stephanie Strassel. 2014. Overview of linguistic resource for the tac kbp 2014 evaluations: Planning, execution, and results. In *Proc. Text Analysis Conference (TAC2014)*.

Jenny Rose Finkel, Trond Grenager, and Christopher Manning. 2005. Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 363–370. Association for Computational Linguistics.

Namrata Godbole, Manja Srinivasaiah, and Steven Skiena. 2007. Large-scale sentiment analysis for news and blogs. *ICWSM*, 7(21):219–222.

Hussam Hamdan, Patrice Bellot, and Frederic Bechet. 2015. Lsislif: Crf and logistic regression for opinion target extraction and sentiment polarity analysis. *SemEval-2015*, pages 753–758.

Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 168–177. ACM.

Long Jiang, Mo Yu, Ming Zhou, Xiaohua Liu, and Tiejun Zhao. 2011. Target-dependent twitter sentiment classification. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 151–160. Association for Computational Linguistics.

Efthymios Kouloumpis, Theresa Wilson, and Johanna D Moore. 2011. Twitter sentiment analysis: The good the bad and the omg! *ICWSM*, 11:538–541.

Himabindu Lakkaraju, Richard Socher, and Chris Manning. 2014. Aspect specific sentiment analysis using hierarchical deep learning. *Advances in Neural Information Processing Systems*.

Yan Li, Yichang Zhang, Xin Tong Doyu Li, Jianlong Wang, Naiche Zuo, Ying Wang, Weiran Xu, Guang Chen, and Jun Guo. 2013. Pris at knowledge base population 2013. In *Proc. TAC 2013 Workshop*.

Margaret Mitchell, Jacqueline Aguilar, Theresa Wilson, and Benjamin Van Durme. 2013. Open domain targeted sentiment. In *Conference on Empirical Methods in Natural Language Processing*, pages 1643–1654.

Kamal Nigam and Matthew Hurst. 2004. Towards a robust metric of opinion. In *AAAI spring symposium on exploring attitude and affect in text*, pages 598–603.

Alexander Pak and Patrick Paroubek. 2010. Twitter as a corpus for sentiment analysis and opinion mining. In *LREc*, volume 10, pages 1320–1326.

Bo Pang and Lillian Lee. 2008. Opinion mining and sentiment analysis. *Foundations and trends in information retrieval*, 2(1-2):1–135.

Maria Pontiki, Haris Papageorgiou, Dimitrios Galanis, Ion Androutsopoulos, John Pavlopoulos, 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.

Maria Pontiki, Dimitrios Galanis, Haris Papageogiou, 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*.

Ellen Riloff and Janyce Wiebe. 2003. Learning extraction patterns for subjective expressions. In *Proceedings of the 2003 conference on Empirical methods in natural language processing*, pages 105–112. Association for Computational Linguistics.

Christina Sauper and Regina Barzilay. 2013. Automatic aggregation by joint modeling of aspects and values. *Journal of Artificial Intelligence Research*.

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. In *Proceedings of the conference on empirical methods in natural language processing (EMNLP)*, volume 1631, page 1642.

Mihai Surdeanu and Heng Ji. 2014. Overview of the english slot filling track at the tac2014 knowledge base population evaluation. In *Proc. Text Analysis Conference (TAC2014)*.

Mihai Surdeanu. 2013. Overview of the tac2013 knowledge base population evaluation: English slot filling and temporal slot filling. In *Proceedings of the Sixth Text Analysis Conference (TAC 2013)*.

Tun Thura Thet, Jin-Cheon Na, and Christopher SG Khoo. 2010. Aspect-based sentiment analysis of movie reviews on discussion boards. *Journal of information science*.

Peter D Turney. 2002. Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 417–424. Association for Computational Linguistics.

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*, pages 347–354. Association for Computational Linguistics.

Jeonghee Yi, Tetsuya Nasukawa, Razvan Bunescu, and Wayne Niblack. 2003. Sentiment analyzer: Extracting sentiments about a given topic using natural language processing techniques. In *Data Mining, 2003. ICDM 2003. Third IEEE International Conference on*, pages 427–434. IEEE.

Hong Yu and Vasileios Hatzivassiloglou. 2003. Towards answering opinion questions: Separating facts from opinions and identifying the polarity of opinion sentences. In *Proceedings of the 2003 conference on Empirical methods in natural language processing*, pages 129–136. Association for Computational Linguistics.

Zhihua Zhang and Man Lan. 2015. Ecnu: Extracting effective features from multiple sequential sentences for target-dependent sentiment analysis in reviews. *SemEval-2015*, page 736.

Lei Zhang and Bing Liu. 2014. Aspect and entity extraction for opinion mining. In *Data mining and knowledge discovery for big data*, pages 1–40. Springer.

Meishan Zhang, Yue Zhang, and Duy-Tin Vo. 2015. Neural networks for open domain targeted sentiment. In *Conference on Empirical Methods in Natural Language Processing (EMNLP 2015)*.

Meishan Zhang, Yue Zhang, and Duy-Tin Vo. 2016. Gated neural networks for targeted sentiment analysis. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, Phoenix, Arizona, USA. Association for the Advancement of Artificial Intelligence*.