Mining Pros and Cons of Actions from Social Media for Decision Support

Ebad Ahmadzadeh
Dept. of Computer Sciences
Florida Institute of Technology
Melbourne, Florida 32901
Email: mahmadzadehe2012@my.fit.edu

Philip K. Chan
Dept. of Computer Sciences
Florida Institute of Technology
Melbourne, Florida 32901
Email: pkc@cs.fit.edu

***Abstract*—The task of mining pros and cons from actions has applications in decision support. Given an action query and social media data, we mine related pros and cons of the action via extracting significant events as potential outcomes of the action. We propose using actions and characteristics to select relevant messages, and adjective vectors to establish similarity among adjectives. We introduce SS to select event headlines, and to rank them in the final pros-and-cons table. Our results on two data sets indicate our algorithm can generate more meaningful pros and cons than an existing algorithm.**

I. Introduction

Discovering pros and cons of actions has many potential applications in decision support, like purchase recommendation, and finding likely side effects of medications. Establishing a knowledge base on actions and outcomes could assist individuals in making decisions by illustrating potential outcomes given an action that they intend to perform. Those outcomes then can be categorized to form pros and cons of the action. In this paper, we propose algorithms to create such a knowledge base of actions and outcomes from social media. Inspired by Kiciman and Richardson [1] we introduce techniques to improve their core components to attain more meaningful pros and cons. Our contributions include:

1) identifying relevant messages containing observations or opinions about the entity of the query by extracting actions and characteristics, as opposed to filtering irrelevant messages in a semi-manual fashion [1],
2) introducing Adjective Vectors to measure semantic similarity between adjectives to improve the clustering quality as in [1],
3) proposing Significance Score (SS) to quantify significance of messages in terms of representing meaningful outcomes, in addition to [1]'s relative likelihood score as a measure to rank distinguishing events, and
4) based on two data sets collected from social media, showing that our algorithm mines more meaningful pros and cons of the given action compared to [1].

We discuss the related work in Sec. II. Sec. III provides the problem statement and describes the different steps of our algorithm. We evaluate our algorithms in Sec. IV and conclude in Sec. V.

II. Related Work

Much research exist based on the assumption that co-occurrence may establish some true relationships between actions and outcomes. For instance, in the health domain, social media studies have found relationships among diseases, medicines, related symptoms and side-effects [2].

Similarity between words can also be measured by vector representation of words. Training of such vectors has been done via different techniques like the ones based on matrix factorization [3] and window-based methods [4]. Pennington et al. [5] propose GloVe, an unsupervised method that benefits from both families and outperforms them on word similarity, word analogy, and name entity recognition tasks.

Kiciman and Richardson [1] investigate the feasibility of mining the relationship between actions and their consequences based on social media. The inputs include a large corpus of personal status messages from social media and an action query. The output is a list of pros and cons of doing the action. One of the main shortcomings of [1] is in the event extraction step where sentences are broken into phrases and then clustered into events. Events consist of short phrases that could be less meaningful sometimes. For example, “damn kitten” or “cat is literally” are phrases from their output table that could not express an outcome without referring to the message they belong to. Therefore, selecting messages that represent the event-phrase seems to be important. However, how to pick the example messages in the pros-and-cons is not clear. Furthermore, they performed semantic correlational analysis to order the events with respect to relative likelihood of the event occurring after doing the action compared to both before doing the action and after doing the reverse action. Although the relative likelihood score captures distinguishing events, the results potentially contain events that are not important consequences. For example, “cat being named” is in the results, but it doesn't seem to be the most significant outcome of adopting a cat.

III. Problem Statement and Algorithm

Given an action, the goal is to discover likely outcomes that could be useful in decision making. The inputs include a large corpus of social media messages, and a query about performing an action. The output is the most likely outcomes of the query action in form of a pros-and-cons table. While we maintain the main skeleton of [1], we propose improvements to its core components. Our algorithm has four main steps that we discuss in the next sections. 1) Find users who performed *Action Query*, and collect a timeline of messages for each user from the *Corpus* (Sec. III-A). 2) Select messages that express **actions** or **characteristics** (Contribution 1) related to the *Action Query* (Sec. III-B). 3) Extract events from messages with techniques including **Adjective Vectors** (Contribution 2; Sec. III-C). 4) Rank the events via Significance Score (SS) (Contribution 3 Sec. III-D).

A. Identifying Relevant Users

From a corpus of social media messages, we find a large number of users who expressed their experience related to the action query. For instance, we search for "adopted a cat|kitty|kitten" when the action query is "adopting a cat". After identifying users who wrote those messages, we collect the entire timeline of messages for each user.

B. Selecting Relevant Messages

The goal of this step is to select messages that describe a relevant situation that the user has experienced. First we use an n-gram approach to filter out non-experiential messages. Then, we select messages with actions and characteristics.

1) Experiential Messages: Relationship between actions and consequences are more meaningful when they are extracted from personal experiences. But social media messages also include other types, like news and advertisements. We use a simplified unsupervised technique based on n-gram models [1] to filter out undesired messages containing certain keywords and phrases.

2) Messages with Actions or Characteristics: Actions and characteristics are usually used in natural languages to express effects and outcomes of an action. Hence, we find messages containing actions or characteristics that refer to the *query entity*. Here we define *query entity* as the main object of the query. For example, given query "adopting a cat", the query entity would be "cat". *Actions* and *characteristics* are represented by verbs and adjectives respectively. An *action* is any verb done by or to the query entity, and similarly, a *characteristic* is any adjective mentioned about the query entity. For instance, given action query "Adopting a cat", and sample message "I love coming home and going to bed because my cute cat cuddles with me.", "cuddle" is an action done by the query entity "cat". Also "cute" is a characteristic about the query entity.

To find messages with such grammatical structures, we extract dependency relationships and part of speech tags from each sentence using Stanford Dependency Parser [6]. Then, we find messages with actions or characteristics via a set of handmade grammar rules: 1) Select any message where the query entity is subject of a verb (action) or where the query entity is subject of a verb and the verb has an adjectival phrase (characteristic). For instance, in "My fat cat is asleep.", "asleep" is the adjectival complement to "is" where "cat" is the subject of "is". 2) select any message with an adjectival modifier for the query entity. For instance, in the previous example, "fat" is an adjectival modifier of the query entity "cat". 3) select any message where the query entity is a object of a verb.

C. Extracting Significant Events

The main goal of this step is to summarize the actions (verbs) and characteristics (adjectives) into events such that each event represents a collection of verbs or adjectives about the same action or characteristic. That is, the verbs and adjectives are clustered (separately) to form events. For example, {"nice", "cute", "lovely"} could form a cluster of adjectives, and {"plays", "runs", "jumps"} could form a cluster of verbs. Event extraction using phrases, as done in previous work [1], provides more effective results than using bag of words as it handles canonicalization. However, it falls short in establishing semantic relationships among words. Since it essentially works based on matching tokens, the clusters are small, with high precision but low recall. As a result, an event can be broken into many small events that otherwise might have formed a significant event. We employ different word representations to establish stronger semantic relationships between verbs and between adjectives. We expect the representations help create clusters with higher recall. Our event extraction algorithm first creates clusters of verbs and adjectives. Then, it identifies a best candidate message and event to represent each cluster.

1) Clustering of Verbs/Adjectives:

Representation of Verbs: We use *Wordnet* [7] hierarchy of verbs. The hierarchy represents different relation types like *is-a, has-a* and it becomes more specific toward leaves. Thus, a data point in this case is a verb token.

Representation of Adjectives: Wordnet does not provide a hierarchy for adjectives, and the task of calculating similarity between adjectives remains difficult in the domain. We propose a different approach based on usage of language by human on the Web where the key assumption is that distribution of similar terms about a characteristic of an entity should be similar. For instance, a particular cat could have a small number of important characteristics such as "cute" that can be expressed by many different words (e.g. "sweet", "lovely"). Those terms are likely to co-occur frequently in comments for the same cat. However, it is unlikely to see terms with opposite meaning (e.g. "ugly") to describe the same cat. Therefore, co-occurrence of similar terms (e.g. "cute" and "nice") are likely to be high on "cute" cats. We use this idea to represent adjectives.

We employ social media posts and comments about the query entity to establish semantic relationships between adjectives (in the experiments we use Reddit (www.reddit.com)

	nice	cute	…	mad
Post1	7	6		1
Post2	18	20		0
Post3	2	0		14
…				
	V_{nice}	V_{cute}		V_{mad}

Figure 1: Adjective vectors: each number shows the occurrence frequency of an adjective in comments to a post.

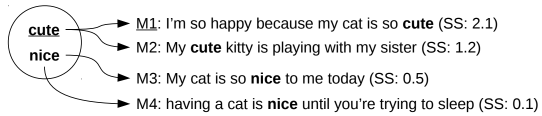

Figure 2: Identifying Representative Messages

data for this purpose). Each post is about an instance of the query entity (e.g. a cat), and the comments discuss the same entity from different perspectives, and mostly express similar characteristics with different words. We represent each post, along with its comments, with one vector. Each vector contains frequency distribution of the adjectives mentioned about the entity in the comments to the post. In other words, each adjective represents a dimension of a post. For example, each row in Fig. 1 is a vector representing a post. Then, each adjective is represented by an *adjective vector* containing the frequency of the adjective in all comments to the original posts. Each column in Fig 1 illustrates an adjective vector. For example, V_{nice} is the adjective vector for "nice", and V_{cute} is the adjective vector for "cute". Since "nice" and "cute" tend to co-occur more frequently than "nice" and "mad" or "cute" and "mad" for the same cat, $V_{nice} - V_{cute} << V_{nice} - V_{mad}$ is likely to hold. That is, the adjective vectors of similar adjectives are expected to be in close proximity. In addition to our proposed adjective vectors, we also use GloVe [5], an unsupervised method for creating word embedding trained on different corpora to represent words.

Distance Function and Clustering Algorithm: We use *hso* [8] for verbs, and cosine for adjectives in *Kmeans++* to cluster verbs and adjectives separately.

2) Identifying Representative Messages and Events: The next step is to identify a single significant message that represents each event cluster. First, each cluster member, whether a verb or an adjective, is associated with one or more messages. Next, we use a scoring mechanism that we refer to as *Significance Score* (SS) to pick an exemplar message for the cluster. Then, that message along with the associated verb or adjective represent the event. Fig 2 illustrates an example cluster with two adjectives ("cute" and "nice"). Each adjective is associated with all messages that contain the adjective. For example, M_1 and $M2$ contain "cute", and M_3 and M_4 contain "nice". Next, we pick the message with the highest significance. The message (e.g. M_1) along with the associated adjective (e.g. "cute") are then selected as the exemplar to represent the cluster.

In order to score messages we employ five factors that contribute to the significance of an event and are sentiment, reasoning, comparison, coverage, and length. We explain each factor in more details. **1) Sentiment Factor:** We use VADER [9], a rule-based sentiment model to calculate sentiment factor as an aggregated score normalized between 0 to 1 to show negative to positive respectively. **2) Reasoning Factor:** Messages with reasoning are highly desired because they provide reasons that probably support their feeling about the outcome. It is calculated as a binary variable that is 1 when any phrase indicating reasoning is observed in the message (e.g. because, therefore, etc.), and it is 0 otherwise. **3) Comparison Factor:** Comparison is often used in decision making process. Comparison factor is also calculated as a binary variable that is 1 when comparison tokens are observed and 0 otherwise. The tokens include both keywords and part-of-speech tags. We use POS tags that are used for comparison words (*JJR, RBR, JJS, RBS*) [10] to identify comparison in sentences. For example, in "cat makes a better pet", the POS tag for "better" is JJR. An advantage of using POS tags is that many words can be represented by one tag. However, POS tagging can be prone to error on incomplete or conversational sentences that usually contain typos. Therefore, we also use a small set of keywords (*more, most, less, enough*). **4) Coverage Factor:** A message is a stronger member of its cluster when it contains more than one cluster word. We define coverage factor as the percentage of cluster words observed in a message normalized between 0 and 1. **5) Length Factor:** Number of words in a message is another indication for a message to be informative. We exclude tokens like urls, hashtags and mentions. This factor is normalized between 0 and 1 by comparing all messages within a cluster. We combine the factors via a weighted sum approach:

$$SS = w_{snt}s_{snt} + w_{res}s_{res} + w_{cmp}s_{cmp} + w_{cov}s_{cov} + w_{len}s_{len} \quad (1)$$

where *SS* is the Significance Score, w_i is the weight used for i^th factor, and s_i is the i^th factor. snt, res, cmp, cov, len represent sentiment, reasoning, comparison, coverage, and length factors respectively. Finally, the message with the highest SS and its associated verb or adjective are selected as the event to represent the cluster to which they belong.

We summarizes the steps of extracting significant events as follows. The inputs are the action verbs and characteristic adjectives along with messages to which they belong, the Reddit data discussed in Sec. III-C1, and cluster sizes for verbs events and adjective events. First, the distance matrix for verbs is created using Wordnet hierarchy of verbs, and then the verbs are clustered to form events. Next, for each verb in the clusters we get all messages from which the verb was extracted as an action verb. Then, we use SS to to select the best message to represent that verb. Subsequently, we create adjective events. First, we create an Adjective Vector for each adjective using Reddit data, as discussed in Sec.

III-C1. Then we follow similar steps as we did for the verbs, namely, calculating distance matrix, clustering, and selecting a representative for each adjective in the clusters. Finally, the verb events and adjective events are returned.

D. Ranking and Categorizing Events

After extracting significant events and finding a representative message for each event, we rank and categorize them into a pros-and-cons table. We follow three steps to generate the table: First, a collection of highly distinguishing events are selected via correlation analysis used in the previous work. Second, the messages from the first step are ranked by our SS. Third, the ranked messages are categorized into pros or cons when their sentiment scores are large enough. Next, we explain each of the three steps in more details.

1) Distinguishing Events via Correlation: We use correlation analysis on preceding and subsequent events after performing the action to infer potential outcomes. This step is equivalent to the correlational analysis with semantic scoring introduced in the previous work [1].

2) Ranking Distinguishing Events: Although distinguishing events ranked by RL are useful, they may not always represent significant events. For example, naming a cat is a distinguishing event, but it is not significant enough to be in the pros-and-cons table. Therefore, we apply our SS to rank events selected from the previous step (Sec. III-C).

3) Categorizing Events: The final step is to categorize the ranked events into two categories of pros and cons. We summarizes the steps for ranking and categorizing events discussed in Sec III-D. The inputs are the events, including both verb and adjective events, weights to calculate SS, the pros-and-cons table size, and the sentiment thresholds for categorizing events into pros and cons. Each event contains a pair of the word (verb or adjective) and the representative message. First, top-k events with highest RL scores are selected. Then the top (distinguishing) events are ranked in decreasing order by SS applied to their messages. Next, sentiment score is calculated for each next event's message from the top of the ranked list. If an event's sentiment score falls in the sentiment threshold conditions (more than +0.5 for pros and less than -0.2 for cons), it is added to the corresponding list. Finally, the pros and cons lists are returned.

IV. Experimental Evaluation

A. Evaluation Criteria

We evaluate effectiveness of the pros and cons extracted by our technique compared to those of the KR15 algorithm [1] that we call KR15 hereafter. Our evaluation criterion is the extent to which events extracted by each algorithm indicate meaningful pros and cons. To establish the ground truth we asked three evaluators who are graduate students in computer science and engineering fields, and are not authors of this paper, to categorize each of the outputted messages into one of three classes {pro, con, neither} based on their personal opinion. Each message's label was then decided based on the majority of the three opinions. To avoid bias toward any algorithm, messages selected by the different algorithms were merged into one set before evaluation. We calculate Discounted Cumulative Gain (DCG) to quantify the ordering quality of the (events) messages in the pros-and-cons table. We expect more relevant messages appear higher in the table. DCG for a pros-and-cons table is calculated as an average between DCG of pros and cons lists.

B. Data

1) Twitter: Two data sets were collected for our experiments based on two action queries: **1.1) Cat Adoption**: We study the consequences of adopting a cat and collect tweets based on search queries such as "adopted a pet", "got a pet", and "got a new pet", where pet is either "cat", "kitty" or "kitten". The query entity is either "cat", "kitty", or "kitten". We collected ~1.8 million tweets from 980 users who adopted a cat in March and May of 2016. For each user, we collected their timeline from three months before and after adoption. **1.2) Buying IPhone 6**: The action query, in this case, is buying an IPhone. We collected ~2.2 million tweets from 1420 users who purchased an IPhone 6 in January and February of 2017. We collected each user's timeline from three months before and after they purchased an IPhone 6.

2) Reddit: We used Reddit data in form of posts and comments about both query entities (cat and IPhone) to train the Adjective Vectors discussed in III-C1. We collected 400 posts about cats and 700 posts about IPhone. Ultimately, we created Adjective Vectors to represent 931 unique adjectives about cat, and 1685 unique adjectives about IPhone.

3) GloVe-Common-Crawl and GloVe-Twitter [1]: In our experiments we also use vectors trained by GloVe [5] as an alternative to our Adjective Vectors.

C. Procedures

We set up our system with four different models for representation of adjectives and verbs: 1) Our Adjective Vectors trained with the Reddit data set for adjectives and Wordnet hierarchies for verbs, 2) GloVe Common Crawl vectors for adjectives and verbs, 3) GloVe Twitter vectors for adjectives and verbs, 4) GloVe Reddit vectors, where we use our Reddit data set to train 200-dimension vectors by the GloVe algorithm. Furthermore, we employ a *hybrid* approach, based on voting among the four models. The voting process affects the event ranking component discussed in III-D2. After selecting the distinguishing events by *RL*, we sort the messages with respect to the number of votes from the four modes in descending order. Next, the messages are selected by SS. The new ordering assigns higher chance of selection to messages with more votes.

[1] https://nlp.stanford.edu/projects/glove/

Table I: DCG of the algorithms on the Cat and IPhone6 data set; IMP is relative improvement over KR15

Algorithm	Cat	IPhone6	Avg	IMP
KR15	1.52	0.91	1.22	—
KR15+SS	2.08	1.93	2.01	65%
AdjectiveVectors+Reddit	2.64	**2.42**	**2.53**	**107%**
GloVe+CommonCrawl	2.44	1.94	2.19	80%
GloVe+Twitter	**2.71**	1.52	2.12	74%
GloVe+Reddit	2.45	2.21	2.33	91%
Hybrid	**2.71**	2.25	2.48	103%

We also implemented the algorithm of Kiciman and Richardson [1] (referred to as KR15 in the results) to be able to compare the results. Since KR15 does not mention a specific way to select the representative message for each event, we assume that it picks a message arbitrarily. But we add another version of this algorithm where we use SS to do the selection. We refer to this version as *KR15+SS* in the results. Moreover, we use the following weights for the factors in SS: ($w_{snt} = 1.0$, $w_{res} = 0.67$, $w_{cmp} = 0.67$, $w_{cov} = 0.1$, $w_{len} = 0.1$)

D. Results on DCG

Table I shows the ranking effectiveness of pros-and-cons based on DCG. In the Cat data, GloVe+Twitter and Hybrid outperform other models. AdjectiveVectors+Reddit stands second. Moreover, it is observed that Hybrid has picked the best ranking among the individual models. In the IPhone6 data set, AdjectiveVectors+Reddit outperforms other models and shows 107% relative improvement over KR15.

1) AdjectiveVectors+Reddit vs GloVe word vectors: Since our four models (AdjectiveVectors+Reddit, GloVe+CommonCrawl, GloVe+Twitter, GloVe+Reddit) all use SS for both selection of the representative messages and ranking of the events, the difference among their DCG scores is mostly due to the difference in precision. For more details about precision results refer to the longer version.

2) AdjectiveVectors+Reddit vs Hybrid: AdjectiveVectors+Reddit outperforms Hybrid in terms of DCG, on average. However, this is not an effect of ranking, because Hybrid shows lower precision than AdjectiveVectors+Reddit on the IPhone6 data.

3) AdjectiveVectors+Reddit vs KR15/KR15+SS: AdjectiveVectors+Reddit outperforms both KR15 and KR15+SS because it uses SS to rank the significant events after selecting the distinguishing ones with RL. The mistakes by AdjectiveVectors+Reddit on the Cat and IPhone6 data occur as high as the second row of cons list. However, KR15 and KR15+SS have more mistakes and they occur as high as the first row.

4) KR15 vs KR15+SS: Although the extracted events and the ranking of those events are the same, our SS finds better representative messages for each event. As a result, many of the mistakes are corrected (20% increase in precision). Since RL is used for ranking of events in KR15+SS, SS can only improve DCG through increasing precision.

E. Examples of Pros-and-Cons tables

1) Cat Adoption: Tables. IIa and IIb illustrate top five pros and cons generated by KR15 and AdjectiveVectors+Reddit on the Cat data respectively. The events are ranked by RL score in KR15. However, they are ranked by SS in AdjectiveVectors+Reddit. Overall, the events extracted by AdjectiveVectors+Reddit represent outcomes of higher quality compared to those extracted by KR15. The only mistake from AdjectiveVectors+Reddit occurs in the second row of cons. The reason to select event *"lazy"* as a con goes back to identifying the representative message for a cluster via SS. *"lazy"* belongs to a cluster of three adjectives {*lazy, obese, fat*}. Looking at the messages within the cluster we find one alternative that could have been picked: *"our fat cat had to be put down. He was just in too much pain."* In this case, the associated event would be "fat". The SS values for the message that appears in our cons list and the alternative message are 3.90 and 3.69 respectively. We observe two main reasons for this undesired selection: First, the sentiment scores of the two messages are -0.57 and -0.51 respectively whereas the alternative message conveys much more negative meaning compared to the selected message. Therefore, the sentiment module [9] is not able to evaluate the alternative message effectively. The second reason that the selected message gains higher SS value is that it contains reasoning token *"because"*.

2) Buying IPhone6: Tables IIc and IId illustrate the top five pros and cons generated by KR15 and AdjectiveVectors+Reddit on the IPhone6 data respectively. Overall, the events extracted by AdjectiveVectors+Reddit represent outcomes of higher quality compared to those extracted by KR15. The reasons for the two mistakes from AdjectiveVectors+Reddit are similar to the ones discussed in IV-E1. They are also discussed in the longer version of this paper.

V. Concluding Remarks

We propose actions (verbs) and characteristics (adjectives) to select relevant messages. Also, we propose adjective vectors to represent adjectives and Wordnet entities to represent verbs. A longer version of this paper is availabe [2].

References

[1] E. Kıcıman and M. Richardson, "Towards decision support and goal achievement: Identifying action-outcome relationships from social media," in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2015, pp. 547–556.

[2] M. Myslín, S.-H. Zhu, W. Chapman, and M. Conway, "Using twitter to examine smoking behavior and perceptions of emerging tobacco products," *Journal of medical Internet research*, vol. 15, no. 8, p. e174, 2013.

[2] http://cs.fit.edu/~pkc/papers/bigdata17long.pdf

Table II: Example Pros & Cons table generated by algorithms: (a) KR15, (b) AdjectiveVectors+Reddit on the Cat data, and (c) KR15, (d) AdjectiveVectors+Reddit on the IPhone6 data. SE represents sentiment score (1=good, -1=bad), and GT (ground truth) illustrates the majority vote on the message by the evaluators (P=pro, C=con, N=neither).

(a) Algorithm: KR15 – Data: Cat

	Pros					Cons				
	Event	Representative Message	SE	RL	GT	Event	Representative Message	SE	RL	GT
1	adorable cat	[...] I've adopted an adorable cat within a span of a month. life is great.	0.81	6.2	P	cat play	My cat doesn't play nice with the dogs [...] so he's in the bedroom for most of the weekend	-0.3	6.1	N
2	my cat is	my cat is literally curious about anything i eat xd.	0.63	5.9	P	ignore me	Accidentally punched my cat in the nose. He's going to ignore me and make me feel guilty [...]	-0.63	5.5	C
3	kitten watched	My kitten watched her namesake get the win and come 1 game away from the.	0.59	5.2	N	kitten is sad	looking back through the window, it seemed my kitten was sad to see me go to work.	-0.45	5.1	N
4	wake up	The entire room just screamed "THE HOUND" and my cat didn't wake up so she's super cool.	0.66	4.8	N	stupid enough	my cat is stupid enough to sleep while eating.	-0.55	4.6	P
5	my kitten is	My kitten is definitely winning.	0.53	4.3	P	tearing up	My cat's tearing up my room trying to kill a fly.	-0.69	4.1	C

(b) Algorithm: AdjectiveVectors+Reddit – Data: Cat

	Pros					Cons				
	Event	Representative Message	SE	SS	GT	Event	Representative Message	SE	SS	GT
1	affectionate	Life is better with a cat. Tigger is affectionate and would make a great lap cat. [...]	0.91	4.94	P	smells	My cat is sleeping in my volleyball bag and I feel bad for him because it smells so bad.	-0.79	4.51	C
2	sweet	I'm so happy [...] that my sweet kitty came back home to me, I missed you so much sweet girl.	0.91	4.93	P	lazy	My cat is so lazy he just dragged himself across my bed because he didn't want to get up om*g.	-0.57	3.90	N
3	cuddles	I love coming home and going to bed because my cat cuddles with me. She is so lovely!	0.88	4.81	P	ignore	Accidentally punched my cat in the nose. He's going to ignore me and make me feel guilty [...]	-0.77	3.52	C
4	mews	Aww. My cat mews so cute. I love him so much.	0.85	4.67	P	claws	My kitten claws my couch and attacks my baby... not so sure I like him anymore.	-0.77	3.43	C
5	hungry	When my cat is hungry, [...] she just puts on her best Im starving face and stares at me.	0.53	3.80	P	mad	My cat is so mad at me being that I took her to the vet today.	-0.63	2.99	C

(c) Algorithm: KR15 – Data: IPhone6

	Pros					Cons				
	Event	Representative Message	SE	RL	GT	Event	Representative Message	SE	RL	GT
1	my new iphone	I'm in love with my new iPhone 6 Plus	0.64	7.7	P	iphone charger	listen! if you have an iphone 6 charger [...] i will literally cry because [...] my phone is dead.	-0.83	6.9	N
2	whip out	gotta whip out the iphone 4 since i got my iphone 6 taken away lmao help me.	0.77	6.8	N	unlock it	[...] i can give you my iphone 6 and i'll unlock it.	-0.88	5.9	N
3	got my new	i got my new phone today [...] still on this iphone 6 cuz i haven't ported my number lol.	0.52	6.2	N	getting my iphone	[...] my sister is 6 and she"s getting my iPhone 6 in two days and has no clue.	-0.78	5.3	C
4	using my iphone	if y'all text me my phone is restoring rn so i'm using my iphone 6 on wifi lmao hit me on here.	0.73	5.5	N	had iphone	my brother [...] got the iphone 6 had it for one day broke it and my mom now got him the 7.	-0.68	3.4	N
5	working perfectly	got my new iphone 6 working perfectly!	0.67	4.5	P	second phone	limited budget it's just a second phone for my kink life. i'm currently using an iphone6.	-0.23	3.1	C

(d) Algorithm: AdjectiveVectors+Reddit – Data: IPhone6

	Pros					Cons				
	Event	Representative Message	SE	SS	GT	Event	Representative Message	SE	SS	GT
1	greatest	[...] my iphone 6 plus was the greatest phone i've ever had. [...]	0.92	6.97	P	trying	hi i'm trying on my iphone6 [...] stuck with an error message at any point i've tried many times!	-0.61	4.03	C
2	turn on	like my old iphone 6 wouldn't turn on and i'm pretty sure it was bc if the last jailbreak i had..	0.79	4.51	N	loses	[...] since updating today my iphone6 loses power rapidly.	-0.61	3.99	C
3	restore	if y'all text me my phone is restoring rn so i'm using my iphone 6 on wifi lm*o hit me on here.	0.73	4.33	N	rebooting	my phone on iphone 6 jailbrake just keeps rebooting my phone randomly. so annoying!	-0.58	3.85	C
4	survived	dropped my naked iPhone 6 into a toilet and it survived so today has been pretty ok.	0.69	4.24	P	went	my iphone 6 went stupid and i can't get an appt at apple till saturday, [...].	-0.53	3.78	C
5	waterproof	my phone just fell in the tub and the music continued to play [...] the iphone6 is waterproof.	0.64	4.06	P	stupid	i love my apple iphone 6plus. but there is no [...] way i'll spend $1k [...] it's just a stupid phone.	-0.86	2.77	C

[3] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman, "Indexing by latent semantic analysis," *Journal of the American society for information science*, vol. 41, no. 6, p. 391, 1990.

[4] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[5] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 1532–1543. [Online]. Available: http://www.aclweb.org/anthology/D14-1162

[6] C. D. Manning, M. Surdeanu, J. Bauer, J. R. Finkel, S. Bethard, and D. McClosky, "The stanford corenlp natural language processing toolkit." in *ACL (System Demonstrations)*, 2014, pp. 55–60.

[7] C. Fellbaum, "Wordnet: an eletronic lexical database. cambridge, massachusetts, eua," 1998.

[8] G. Hirst, D. St-Onge *et al.*, "Lexical chains as representations of context for the detection and correction of malapropisms," *WordNet: An electronic lexical database*, vol. 305, pp. 305–332, 1998.

[9] C. J. Hutto and E. Gilbert, "Vader: A parsimonious rule-based model for sentiment analysis of social media text," in *Eighth international AAAI conference on weblogs and social media*, 2014.

[10] N. Jindal and B. Liu, "Mining comparative sentences and relations," in *AAAI*, vol. 22, 2006, pp. 1331–1336.