Construction and Performance Analysis of a Groomed Polarity Lexicon Derived from Product Review Source Datasets

Derek Colley
School of Digital, Technologies and Arts
Staffordshire University
Stoke-on-Trent, United Kingdom
derek.colley@staffs.ac.uk

Md Asaduzzaman
School of Digital, Technologies and Arts
Staffordshire University
Stoke-on-Trent, United Kingdom
md.asaduzzaman@staffs.ac.uk

***Abstract*—Using a large, publicly-available dataset [1], we extract over 51 million product reviews. We split and associate each word of each review comment with the review score and store the resulting 3.7 billion word- and score pairs in a relational database. We cleanse the data, grooming the dataset against a standard English dictionary, and create an aggregation model based on word count distributions across review scores. This renders a model dataset of words, each associated with an overall positive or negative polarity sentiment score based on star rating which we correct and normalise across the set. To test the efficacy of the dataset for sentiment classification, we ingest a secondary cross-domain public dataset containing freeform text data and perform sentiment analysis against this dataset. We then compare our model performance against human classification performance by enlisting human volunteers to rate the same data samples. We find our model emulates human judgement reasonably well, reaching correct conclusions in 56% of cases, albeit with significant variance when classifying at a coarse grain. At the fine grain, we find our model can track human judgement to within a 7% margin for some cases. We consider potential improvements to our method and further applications, and the limitations of the lexicon-based approach in cross-domain, big data environments.**

Keywords—Sentiment analysis, natural language processing, relational databases, language analysis, SQL aggregation.

I. Introduction

Sentiment analysis involves the assessment of natural language text segments to determine the degree of membership within some nominal classification taxonomy. This can include, for example, assessment of social media posts to determine whether any given post is positive or negative in tone. This application of sentiment analysis can be valuable to businesses and organisations looking to extend their customer management processes to route product or service complaints to the relevant handlers; to establish consumer attitudes towards a brand; or to augment existing customer segmentation strategies.

Challenges in this space include improving the classification success against non-standard phrase forms and short text segments (such as Twitter posts). In this paper, we propose the derivation of a sentiment lexicon from an existing dataset of customer product reviews, and we present, test and validate a method for text segment deconstruction and sentiment calculation against the derived dataset.

II. Related Work

Sentiment analysis is the art of programmatically extracting meaning from often abstract and unstructured segments of text. Such analysis is useful in crossing the divide between quantitative, empirical classification and qualitative codification, and the outputs of accurate sentiment analysis have a host of potentially useful applications. These include inclusion in a branching strategy for efficient customer touchpoint handling in social media contexts, the provision of management information on brand perception in the marketplace [2], helping to displace risk of reputational damage, use in determining adverse drug effects from medical message boards and more [3, 4, 5]. Sentiment analysis has been used to classify human communication content as objective or subjective [6]; the objective being the facts of the matter at hand and the subjective being the channel of communication carrying sentiment, and further noted that accurate textual sentiment analysis is an unsolved problem. Not all human communication carries emotional meaning, depending on the context; opinions are more emotionally loaded, for example, than dry factual reporting, but as language evolves, the challenges in meaning extraction become progressively more difficult. This field is a subset of natural language processing (NLP) problems and termed Sentic computing.

Algorithmic approaches to addressing sentiment extraction differ widely. These include supervised and unsupervised machine learning methods, lexicon-based methods, use of keywords and concept extraction [7]. Our research focuses on sentiment polarity extraction using the lexicon-based method, where opinions are held to be positive, negative or somewhere on a bounded scale between these two finite extremes [8]. We attempt this not by using a dictionary-based approach, but by creating a weighted lexicon model, where each word in the lexicon is assigned a score, then applying this model to new text inputs.

While this approach is similar to unsupervised machine learning in neural networks, the fundamental mechanism is instead algorithmic in nature and founded on a very large training data set. Weights are set using simple aggregate functions rather than complex feed-forward networks. Similar approaches have been used before; Phrasal, rather than word, lexicon weighting has been used in an adaption of Turney's algorithm [9]; an earlier study [10] presents a series of information retrieval weighting schemes for sentiment analysis, albeit mostly, but not exclusively, centred on machine learning techniques such as the Support Vector Machine (SVM). These individual frameworks have varying degrees of success and are in many cases specifically tuned to known data sets.

We argue that, given a sufficiently-large corpus of balanced data, an algorithmic approach of simple summation and range normalisation can provide performance advantages through lower complexity when a model derived from this approach is used as a quantitative classification mechanism. This is not to denigrate SVM and other ML methods; SVMs [10] are ubiquitous in part because they are able to deal with categorical, hierarchical, semi-structured and ordinal data, whereas simple functions have a brittleness which suits them to low-dimensional, atomic, structured datasets. This criticism has been extended [11], noting that the lexicon-based method is vulnerable to bias in the data source and cannot adapt well to unstructured data sources, albeit also noting the significant performance gain of this method over others. We directly observed this bias in our experiments, with a baseline bias swing of more than +0.30 from mean average against a sample of over 51 million records with more than 3 billion words, explained entirely by the source of the data.

Other challenges in lexicon-based sentiment analysis present themselves. These include the use of non-standard language artefacts such as emojis, abbreviations, misspellings, shortenings of common words and slang. In filtering against a standard English dictionary, potentially interesting sentiment information is lost; an active area of research addressed by, amongst others, For example, an emoji lexicon has been developed in mitigation to this issue [12]. The evidence in the literature indicates many issues in Sentic computing remain research challenges, including the difficulties of separating fact and opinion within datasets [13], or subjectivity detection, noting particular difficulty in classifying weakly-subjective sentences, and a recent survey analysed 47 previous studies in sentiment analysis [14] noting that certain factors such as world (or domain) knowledge, 'bi-polar' words and large lexicons have a deleterious effect on overall accuracy rates.

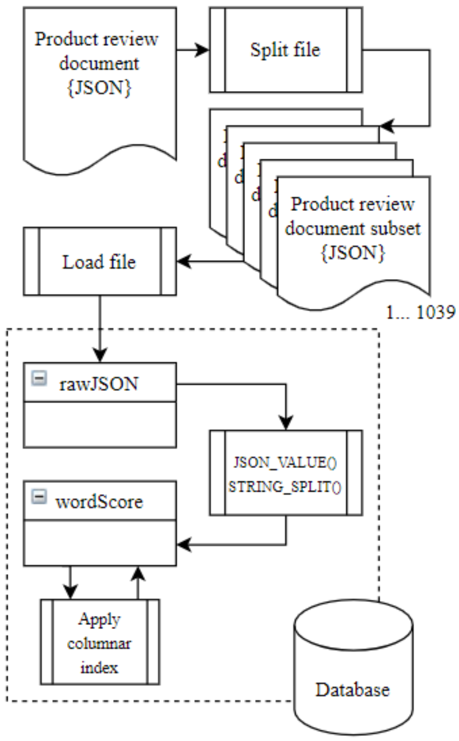
Nevertheless, lexicon-based sentiment analysis has a strong theoretical and practical basis. In a dated but seminal paper, an algorithmic application of a novel lexicon analysis method termed 'SO-CAL' was described. This verified the accuracy of the polarity classification through cross-tabulation with human participant classifications and showed it was able to perform consistently and across different domains of data [15]. This example is one of many case studies in the literature. Our contribution to this field illustrates our outcomes from building and testing a simple unsupervised sentiment polarity calculator applied to a large and unstructured data set, and the applicability of this set to another domain, with the aim of establishing the limitations of this approach in a modern big data setting.

III. Creating the Sentiment Dictionary

A. Sourcing the ratings data

The ratings data is supplied as a publicly accessible dataset, described in the literature [1, 16], and consists of a record set of 51m Amazon product reviews for books, dating to 2018. We downloaded this data, compromised of a single compressed file in JSON format, and examined the schema. In order to work with the file it was necessary to split it into smaller pieces for data ingestion; we wrote a simple file splitter program in Python to do this, resulting in 1,036 files of approximately 30-50MB per file in size. Each except the last file contained exactly 50,000 documents. Next, we iterated through each file, loading each document into Microsoft SQL Server as a single NVARCHAR(MAX) uniquely identified by a auto-sequence number and timestamp. This resulted in 51m records in SQL server, each numbered. Using the JSON_VALUE() SQL function and the STRING_SPLIT() SQL function nested in a common table expression, we then iterated through the 51m records in batches of 100,000 and extracted the *overall* (review score) and *reviewText* (customer review) elements from each JSON document, splitting *reviewText* into its component words (space-separated) and storing the *overall* numeric, renamed to *score*, alongside each word, renamed to *word*. This resulted in 3,795,765,817 rows in total, each row containing a single *score* and a single *word*, each pair representing an occurrence of the word in the whole review dataset and the associated parent review score. For performance improvements when reading the data for the next steps, we converted the rowstore heap table to a columnar indexed table. Fig. 1 illustrates the preparation process.

Fig. 1. Data preparation process.



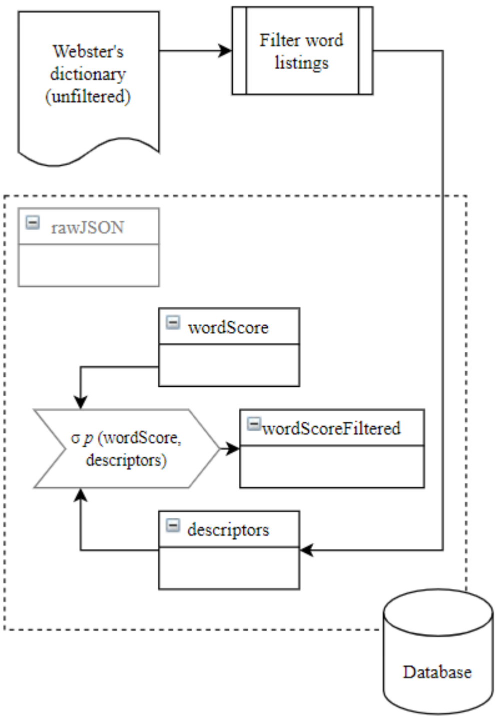
B. Data cleansing

We required a reasonably complete English dictionary upon which we could cleanse our collected data of misspellings, orphaned punctuation marks and other errata. We chose Webster's Unabridged Dictionary (2009) hosted at Project Gutenberg [17] which has a plain-text UTF-8 version available for download. Words are provided in uppercase on their own line with variants of the word separated by semicolons; descriptive data is given in mixedcase. Given this structure it was straightforward to read this word data and extract from the dictionary, storing them in a single-column SQL table. This provided 102,668 distinct words. We recognised there are various forms of word (e.g. 'wonderful', 'wonderfully') and various verbs that may or may not carry sentiment; however, given the complexity of the English language we did not attempt to augment our list through algorithmic extension of words but rather relied on the list of variants within the reference dictionary to provide near-complete coverage. We acknowledge this may have led to the inadvertent exclusion of valid variant forms and the difficulty of English language categorisation is discussed further in our Conclusions.

We filtered our list of 3,795,765,817 words to a shortened list of 2,925,874,997 words by selecting all matching words against the Websters dictionary table into a new table using an inner (predicated) join.

Next, given that case is irrelevant to sentiment, we updated all entries in our table to uppercase format, although this is not strictly necessary as the collation of our database was case-insensitive. This resulted in a table with rating score ('score') and valid descriptor ('word'). Fig. 2 illustrates the data cleansing process.

FIG. 2. DATA CLEANSING PROCESS.



C. Aggregation and score normalisation

With a dataset containing one entry for every occurrence of a word and its accompanying score (for the parent review as a whole), we transformed the data into an aggregate grouping over the word column, calculating both the $\bar{x}$ score (per word) and the count of words. This resulted in a new set of distinct words with all duplicates removed, together with average score and frequency count.

To normalise the range of scores, we calculated the final sentiment score x in the range 0-1 using an ordinary normalisation function. The expressions min(X) and max(X) resolve to 1 and 5 respectively, the boundary of the rating score. Eq. 1 shows this function.

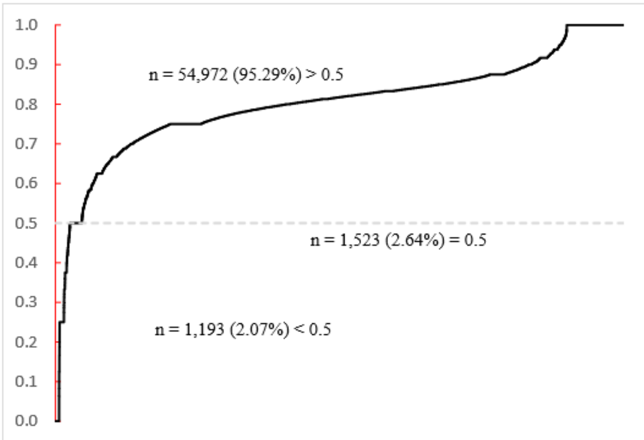
$$f(x) = \left(\frac{x - \min(X)}{\max(X) - \min(X)} \right), \forall x \text{ in } X \qquad (1)$$

This resulted in a set of distinct words and associated normalized sentiment scores which we round to 3 d.p., representing a single scalar sentiment score $0 <= x <= 1$ for each word. At this point, the original ratings are discarded.

This process leaves a total of 57,688 distinct words, each with a single normalised sentiment score in the range 0-1, where 0 represents negative and 1 represents positive sentiments.

We note a potential issue; the resulting range of values is not uniformly distributed. Instead, we find 95.29% of all scores claim a positive sentiment (above the neutral value of 0.5); 2.64% of all scores are exactly neutral at 0.5; the remaining 2.07% of all scores claim a negative sentiment (below the neutral value of 0.5). This is indicative of the parent reviews, where similar percentages are rated 3, 4 or 5 'stars'. The consequence of this skew is that negative-sentiment words may be unrepresented. This skew, identical to the *z*-score distribution except bounded between 0.0-1.0, is shown in Fig. 3.

FIG. 3. SCORE DISTRIBUTION (NORMALISED)



We overcome this issue by resetting the median point at which we define a 'negative' or a 'positive' sentiment to the median of the normalised scores, which is 0.817 for our lexicon. In doing so, we rebalance the population of words on either side of this dividing line. The function for midpoint correction for all scores s is thus:

$$\begin{aligned} f(s) &= 0.817 - s + 0.5 \therefore \\ f(s) &= 1.317 - s \end{aligned} \qquad (2)$$

This function (2) uses the difference between the midpoint (0.5) and the median (0.817), equalling 0.317, adds this value to 1 and subtracts the score s from this number, resulting in a median-adjusted s score for all values s in this dataset. Next, we demonstrate the creation of a user-defined function to take advantage of this lexicon for sentiment analysis.

D. Programmatic Access

To implement a mechanism to access the data programmatically, we created a SQL user-defined function (UDF), returning the sentiment score (a numeric x bounded to $0 <= x <= 1$ to 3 d.p.) given a string expression e of input words with no upper bound on length. This function can be called in the FROM clause using any regular SQL expression; in the relational algebra, it forms a new relation R with no predicate conditions and is join- and union-compatible to other relations. We make use again of the STRING_SPLIT() SQL function and use $\bar{x}$ smoothing to calculate the similarity score for any given expression e.

IV. Testing and Validation

To test the efficacy of our implementation, we sourced Twitter data from the Internet Archive [18], extracting all tweets for a single arbitrarily-chosen day. We considered all tweets with substantial English-language content (defined as at least 8 separate words) in the *'$.extended_tweet.full_text'* field (if exists, per record), and extracted 100 English-language tweets from the set, filtering out for pornographic and offensive content.

We anonymised the metadata and removed identifiable information, such as user handles, then prepared this set as a survey instrument which we administered to 22 participants. Each participant was given the same survey and asked to read and rate each of 100 tweets for negative, neutral or positive sentiment using a ten-point scale. We then calculated the average score given by each participant across each tweet to set a sentiment score which we treat as a truthful reflection of the tweet sentiment and divided by 10 to normalise. We prepared the same ordered set of 100 tweets and applied our UDF function against them, setting the neutral point at the calculated balance point of 0.817 (see Eq. 2), resulting in 100 sentiment scores which we annotated with the appropriate classification. Table 1 illustrates this data.

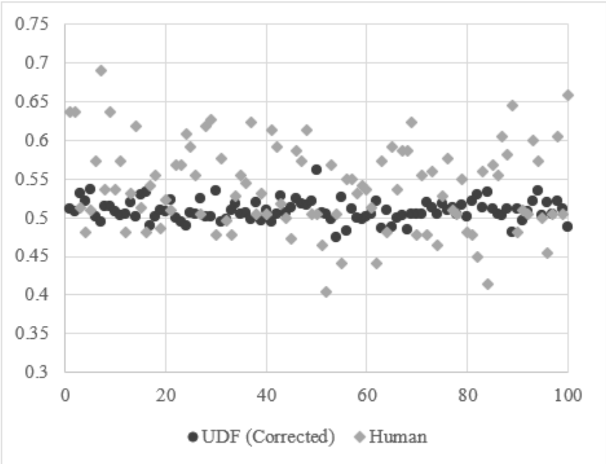
Table I. Average sentiment scores – UDF vs. Human

Tweet #	UDF ($\bar{x}$)	Human ($\bar{x}$)	UDF Δ	Tweet #	UDF ($\bar{x}$)	Human ($\bar{x}$)	UDF Δ
Tweet 1	0.511	0.636	-0.125	Tweet 51	0.506	0.464	0.042
Tweet 2	0.508	0.636	-0.128	Tweet 52	0.504	0.405	0.099
Tweet 3	0.532	0.514	0.018	Tweet 53	0.498	0.568	-0.070
Tweet 4	0.522	0.482	0.040	Tweet 54	0.475	0.505	-0.030
Tweet 5	0.537	0.509	0.028	Tweet 55	0.526	0.441	0.085
Tweet 6	0.501	0.573	-0.072	Tweet 56	0.482	0.550	-0.068
Tweet 7	0.495	0.691	-0.196	Tweet 57	0.511	0.550	-0.039
Tweet 8	0.514	0.536	-0.022	Tweet 58	0.499	0.532	-0.033
Tweet 9	0.514	0.636	-0.122	Tweet 59	0.498	0.541	-0.043
Tweet 10	0.508	0.536	-0.028	Tweet 60	0.502	0.536	-0.034
Tweet 11	0.503	0.573	-0.070	Tweet 61	0.505	0.514	-0.009
Tweet 12	0.504	0.482	0.022	Tweet 62	0.521	0.441	0.080
Tweet 13	0.519	0.532	-0.013	Tweet 63	0.486	0.573	-0.087
Tweet 14	0.502	0.618	-0.116	Tweet 64	0.510	0.482	0.028
Tweet 15	0.529	0.514	0.015	Tweet 65	0.488	0.591	-0.103
Tweet 16	0.533	0.482	0.051	Tweet 66	0.500	0.536	-0.036
Tweet 17	0.490	0.541	-0.051	Tweet 67	0.503	0.586	-0.083
Tweet 18	0.502	0.555	-0.053	Tweet 68	0.484	0.586	-0.102
Tweet 19	0.509	0.486	0.023	Tweet 69	0.505	0.623	-0.118
Tweet 20	0.508	0.523	-0.015	Tweet 70	0.504	0.477	0.027
Tweet 21	0.523	0.509	0.014	Tweet 71	0.504	0.555	-0.051
Tweet 22	0.500	0.568	-0.068	Tweet 72	0.519	0.477	0.042
Tweet 23	0.494	0.568	-0.074	Tweet 73	0.513	0.559	-0.046
Tweet 24	0.490	0.609	-0.119	Tweet 74	0.504	0.464	0.040
Tweet 25	0.507	0.591	-0.084	Tweet 75	0.518	0.527	-0.009
Tweet 26	0.504	0.555	-0.051	Tweet 76	0.509	0.577	-0.068
Tweet 27	0.524	0.505	0.019	Tweet 77	0.513	0.509	0.004
Tweet 28	0.501	0.618	-0.117	Tweet 78	0.505	0.505	0.000
Tweet 29	0.501	0.627	-0.126	Tweet 79	0.516	0.550	-0.034
Tweet 30	0.535	0.477	0.058	Tweet 80	0.502	0.482	0.020
Tweet 31	0.495	0.577	-0.082	Tweet 81	0.522	0.477	0.045
Tweet 32	0.497	0.495	0.002	Tweet 82	0.530	0.450	0.080
Tweet 33	0.510	0.477	0.033	Tweet 83	0.513	0.559	-0.046
Tweet 34	0.518	0.527	-0.009	Tweet 84	0.533	0.414	0.119
Tweet 35	0.505	0.555	-0.050	Tweet 85	0.512	0.568	-0.056
Tweet 36	0.507	0.545	-0.038	Tweet 86	0.505	0.555	-0.050
Tweet 37	0.497	0.623	-0.126	Tweet 87	0.503	0.605	-0.102
Tweet 38	0.520	0.505	0.015	Tweet 88	0.512	0.582	-0.070
Tweet 39	0.496	0.532	-0.036	Tweet 89	0.481	0.645	-0.164
Tweet 40	0.510	0.505	0.005	Tweet 90	0.512	0.482	0.030
Tweet 41	0.495	0.614	-0.119	Tweet 91	0.496	0.509	-0.013
Tweet 42	0.504	0.591	-0.087	Tweet 92	0.508	0.505	0.003
Tweet 43	0.528	0.518	0.010	Tweet 93	0.521	0.600	-0.079
Tweet 44	0.507	0.500	0.007	Tweet 94	0.534	0.573	-0.039
Tweet 45	0.513	0.473	0.040	Tweet 95	0.503	0.500	0.003
Tweet 46	0.524	0.586	-0.062	Tweet 96	0.519	0.455	0.064
Tweet 47	0.518	0.573	-0.055	Tweet 97	0.504	0.505	-0.001
Tweet 48	0.517	0.614	-0.097	Tweet 98	0.522	0.605	-0.083
Tweet 49	0.521	0.505	0.016	Tweet 99	0.511	0.505	0.006
Tweet 50	0.561	0.505	0.056	Tweet 100	0.487	0.659	-0.172

We observe that our UDF function is notably cautious in score allocation, with a score range of 0.086, a maximum score of 0.561 and a minimum score of 0.475. Compare this against the human participant results with a range of 0.286, maximum of 0.691 and minimum of 0.4; a range increase of 332%.

To establish the extent of correlation between the UDF-generated scores and the human-generated scores, we first chart the data points on a scatter diagram (as shown in Fig. 4). We then calculate the bivariate correlation co-efficient (PCC) in the normal way as 0.39, illustrating a weak but present correlation between the two variable sets.

FIG. 4. COMPARISON OF UDF- VS. HUMAN SENTIMENT SCORES, PER TWEET



Another way we may establish relationships is by considering the midpoint 0.5 as the neutral point and classifying all scores above 0.5 as positive and all scores below 0.5 as negative regardless of fine-grained classification. Thus, we obtain the results shown in Table 2.

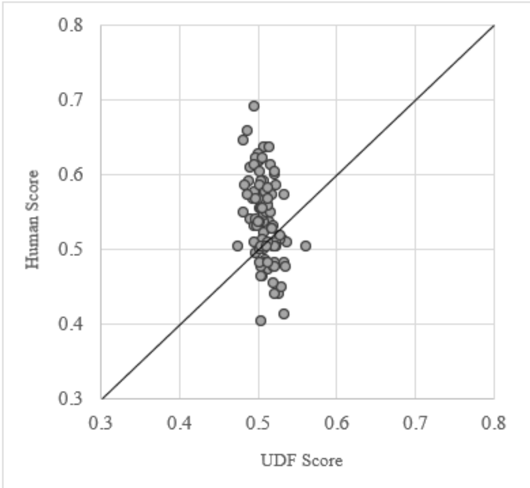
TABLE 2. UDF VS. HUMAN SCORE PERFORMANCE METRICS

UDF: Total positive tweets (s > 0.5)	78
UDF: Total neutral tweets (s = 0.5)	0
UDF: Total negative tweets (s < 0.5)	22
Human: Total positive tweets (s > 0.5)	76
Human: Total neutral tweets (s = 0.5)	0
Human: Total negative tweets (s < 0.5)	24
UDF vs. human s-score agreement	56%
UDF vs. human s-score disagreement	44%

We can observe that the overall volume of classification was similar for both UDF vs. human performance – the UDF classified 78 tweets as positive vs. human classification of 76 tweets as positive, and likewise negative; however, there was significant variance on a per-item level. In 56% of cases there was agreement on positive vs. negative between UDF and human scores. The UDF classifies very near the 0.5 midpoint boundary and the variance inequality means there is significantly more room for individual classification error.

Another way of visualising the difference of our results is to chart the human-determined sentiment scores against the UDF-determined sentiment scores pairwise as Cartesian co-ordinates. The ideal line follows x = y; in this scenario, human-moderated scores would match UDF-generated scores. This is shown in Fig. 5.

FIG. 5. HUMAN VS UDF-GENERATED SCORES, SHOWN ON THE PLANE



We can see, as per our PCC correlation calculation of 0.39, there is a weak but present relationship between human- and UDF-driven score generation using our lexicon and method; that in general, the UDF method is conservative in range and consequently the outcome of text classification using this data source and this method yields a 6% advantage over random chance when using coarse, or discrete, classification.

Finally, we can examine the average difference between human- and UDF-driven scoring by examining the deltas between the mean of the human scores and the UDF-generated score, per text item. These deltas were illustrated in Table 1. We note the average difference between them is just -0.031, with a range of 0.315 (31.5% potential swing) and a low standard deviation of +/-6.3% (0.063, bounded from 0.0-1.0) meaning that UDF-driven scoring generally tracks human-driven scoring to within a 7% margin when considering a continuous score range.

V. CONCLUSIONS AND FUTURE WORK

It is evident from the literature review that Sentic computing continues to present challenges for researchers and industry practitioners in ensuring accuracy in the face of the ever-changing variety of data available to mine, the constant evolution of language to incorporate new vocabulary and phrasings, the difficulty of classifying words, phrases and larger text blocks into quantised groupings and, not least, the difficulties in telling fact from fiction. In this research, we investigated lexicon-based sentiment analysis using a polarity weighting technique and applied this to a big data set derived from Amazon product reviews to create a training dataset; we then attempted to use this training set against another domain of data, a selection of random posts on a social media platform. We found a middling degree of success when compared against human performance at the same task, with 56% of posts classified correctly by the algorithm; we found a better degree of accuracy when looking at particular cases, tracking to within 7% of human performance. We conclude that although lexicon-based methods are applicable to big data sets, as evidenced in our case study, there remain challenges to be solved in applying such rigid polarity lexicons across domains, and these challenges are exacerbated by different vocabularies, intentions and inconsistencies naturally present within informal human language.

REFERENCES

[1] McAuley, J. "Amazon product data", 2018. Available at: http://jmcauley.ucsd.edu/data/amazon/ [Accessed 22 Apr. 2021].

[2] Kaiser, C., Schlick, S. and Bodendorf, F., “Warning system for online market research - Identifying critical situations in online opinion formation”. In Journal of Knowledge-Based Systems, vol. 24, issue 6, pp. 824-836, August 2011.

[3] Al-Moslmi, T., Omar, N., Abdullah, S. and Albared, M, “Approaches to cross-domain sentiment analysis: A systematic literature review”. In IEEE Access, vol. 5, pp.16173-16192, 2017.

[4] Kumar, A. and Jaiswal, A., “Systematic literature review of sentiment analysis on Twitter using soft computing techniques”. In Concurrency and Computation: Practice and Experience, vol. 32, issue 1, pp. 5107. 2020.

[5] Mohammad, S.M., “Challenges in sentiment analysis. In A practical guide to sentiment analysis”, pp. 61-83, 2017. Springer.

[6] Nanli, Z., Ping, Z., Weiguo, L.I. and Meng, C., “Sentiment analysis: A literature review”. 2012 International Symposium on Management of Technology (ISMOT), pp. 572-576, November 2012. IEEE.

[7] Qazi, A., Raj, R.G., Hardaker, G. and Standing, C., “A systematic literature review on opinion types and sentiment analysis techniques”. In Internet Research, vol. 27, issue 3, June 2017.

[8] Cambria, E. and White, B., “Jumping NLP curves: a review of natural language processing research. In IEEE Computational Intelligence Magazine, Vol. 9, issue 2, pp. 48-57. 2014.

[9] Das, A., “Sentiment analysis”. 2017 8th International Conference on Computing, Communication and Networking Technologies (ICCCNT), pp. 1-6, August 2017. IEEE.

[10] Paltoglou, G. and Thelwall, M., “A study of information retrieval weighting schemes for sentiment analysis”. Proceedings of the 48th annual meeting of the association for computational linguistics, pp. 1386-1395, July 2010.

[11] Shayaa, S., Jaafar, N.I., Bahri, S., Sulaiman, A., Wai, P.S., Chung, Y.W., Piprani, A.Z. and Al-Garadi, M.A, “Sentiment analysis of big data: methods, applications, and open challenges”. In IEEE Access, vol. 6, pp.37807-37827, 2018. IEEE.

[12] Fernández-Gavilanes, M., Juncal-Martínez, J., García-Méndez, S., Costa-Montenegro, E. and González-Castaño, F.J., “Creating emoji lexica from unsupervised sentiment analysis of their descriptions”. In Expert Systems with Applications, vol. 103, pp.74-91, 2018.

[13] Chaturvedi, I., Cambria, E., Welsch, R.E. and Herrera, F., “Distinguishing between facts and opinions for sentiment analysis: Survey and challenges”. In Information Fusion, vol. 44, pp.65-77, 2018.

[14] Hussein, D.M.E.D.M., “A survey on sentiment analysis challenges”. In Journal of King Saud University-Engineering Sciences, vol. 30, issue 4, pp.330-338, 2018.

[15] Taboada, M., Brooke, J., Tofiloski, M., Voll, K. and Stede, M., “Lexicon-based methods for sentiment analysis”. In Computational linguistics, vol. 37, issue 2, pp.267-307, 2011.

[16] J. Ni, J. Li and J. McAuley, “Justifying recommendations using distantly-labeled reviews and fine-grained aspects”, in Empirical Methods in Natural Language Processing (EMNLP), 2019. Available at: http://cseweb.ucsd.edu/~jmcauley/pdfs/emnlp19a.pdf [Accessed 22 Apr. 2021].

[17] Project Gutenberg, “Webster’s Unabridged Dictionary by Various”, 2009. Available at: http://www.gutenberg.org/ebooks/29765 [Accessed 22 Apr. 2021].

[18] Internet Archive, “Archive Team Twitter Stream 2020-10”, 2020. Available at: https://archive.org/details/archiveteam-twitter-stream-2020-10 [Accessed 23 Apr. 2021].