LexScore: A Semantic Approach to Scoring Domain Specific Sentiment Lexicons

Shantanu Kumar
inFeedo
4864/24, Jain Bhawan
Daryaganj, Delhi 110002
shantanu@infeedo.com

Abstract

The sentiment of a word varies based on its context of usage: the words used around it and the part-of-speech it is used as. This paper proposes a technique to suggest the sentiment of a word by combining its part-of-speech and the semantic similarities of its co-occurrences with both context-specific and pre-trained embeddings to achieve powerful and fast results. A study was conducted across domains and sub-domains to measure variance of sentiment by switching domains and switching context within the same domain. Re-scoring a commonly used polarity lexicon showed that 10% of words changed scores while switching domains and 8% changed scores within domains while switching context. Part of Speech analysis on 65,353 commonly used sentiment lexicons showed that 81% of sentiment bearing (non-neutral) lexicons were of the tags NN (Common Noun), JJ (Adjective) or NNS (Proper Noun).

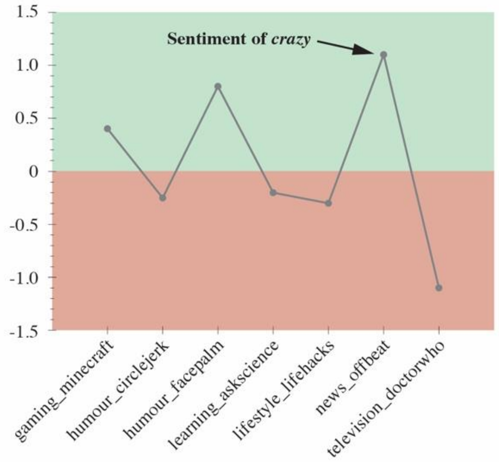

Figure 1: The sentiment scores of *crazy* in different contexts within reddit communities computed using LexScore.

1 Introduction

Assigning sentiment scores to domain and context-specific words for lexicon-based sentiment analysis requires immense domain knowledge. The word *crazy* would have a negative connotation in a scientific setup whereas it would be considered positive in the informal news domain (Figure 1). Understanding the overarching theme of the sentiment of *crazy* with respect to the domain is relatively easier; although the magnitude of the sentiment score to assign still remains a task to calculate owing to the many contexts of usage of the word within the same domain and most times, expensive and time-consuming (Mohammad and Turney, 2010; Fast et al., 2016) since it may require domain experts to work together with annotators; which also allows room for bias based on domain-specific usage or demographic variation (Deng et al., 2014; Hovy, 2015; Yang and Eisenstien, 2015). Large scale web-derived lexicon generation techniques (Velikovich et al., 2010) do not cater well to variation in context of usage and may introduce bias which may further impact the study of opinion and research.

To address these requirements, the paper introduces `LexScore:` an algorithm that learns from domain-specific corpora and enhances pre-existing sentiment lexicons. *LexScore* combines a widely used method of using co-occurrences with word embeddings and part of speech tags to generate sentiment scores for input words. The algorithm is designed to maintain baseline accuracies while building heavily on speed to cater to both researchers and industry implementations where infrastructure spends are limited.

The key contributions of this work are:

1. A novel sentiment scoring algorithm combining word embeddings from domain-specific corpora and part-of-speech tags.
2. The re-scored Jockers (2015b) and Rinker's augmented Hu & Liu (2004) polarity lexicons (Rinker, 2015a) to cater to 51 contexts across 7 domains.

This work aims to collate multiple ways of scoring sentiment-bearing lexicons into one algorithm, maintaining speed and scalability. The re-scored lexicons along with code for the pre-processing and the algorithm will be made available in an R package.

2 Related Work

This work builds upon a wealth of previous opinion papers and research on inducing and assigning scores to sentiment lexicons. Often, the two categories being followed for this task are corpus-based (Hatzivassiloglou and McKeown, 1997; Wiebe, 2000; Turney and Littman, 2003; Kanayama and Nasukawa, 2006) and relying on co-occurrence (Igo and Riloff, 2009; Riloff and Shepherd, 1997; Turney and Littman, 2003). Recent work involves construction of lexical graphs and label propagation (Hamilton et al., 2016).

This work also takes into account the implicit consensus that dictionary-based approaches generate higher-quality lexicons due to their generation requiring domain experts and being cleaner (Hamilton et al., 2016) – the aim is to combine both categories but would fall in the category of dictionary-based approaches.

There is work that shows different POS categories contribute to sentiment in varying degrees and uses Maximum Entropy Modelling to achieve results (Nicholls and Song, 2009). In the interest of speed and scalability of the algorithm, the paper introduces a vanilla filtering approach for words.

Existing methods of generating lexicons primarily cater to sentence-based sentiment analysis (i.e., they provide weighted sentiments only per sentence, e.g., the *R*- package *sentimentr*, Rinker, 2015b). This paper aims to draw out sentiment scores for each word and not just the ones that would normally be sentiment-bearing. For example, "Woah crap that car is amazing!" has a positive connotation but the sentiment bearing words here (crap, amazing) cancel each other out in effect, resulting in a neutral sentiment. The aim is to reassign a positive score to "crap" to capture the true sentiment of this word in the sentence.

This work is inspired by (Hamilton et al., 2016; Nicholls and Song, 2009; Qiu et al., 2009) but aims to achieve similar results using a novel approach that gives importance to speed and interpretability.

3 Polarity Assignment

The proposed algorithm first initializes pre-trained embeddings or builds domain-specific embeddings to have high quality semantic representations for words using a vector space model and then constructs a term-co-occurrence matrix from a tokenized corpus using a skip-gram model that calculates weights given by the formula

$$\text{weight} = (1/\text{offset}), (1)$$

where the $offset$ is the distance of each word from the current word (text2vec.org – Selivanov and Wang, 2016; Pennington et al., 2014) with varying window sizes based on the size of the corpus and the accuracy/speed tradeoff. A pre-existing sentiment lexicon is then used as input to re-create for the specific domain. The algorithm is created to address three key requirements:

1. **Replicable:** The algorithm takes inputs as pre-existing lexicons that are re-scored according to the domain.

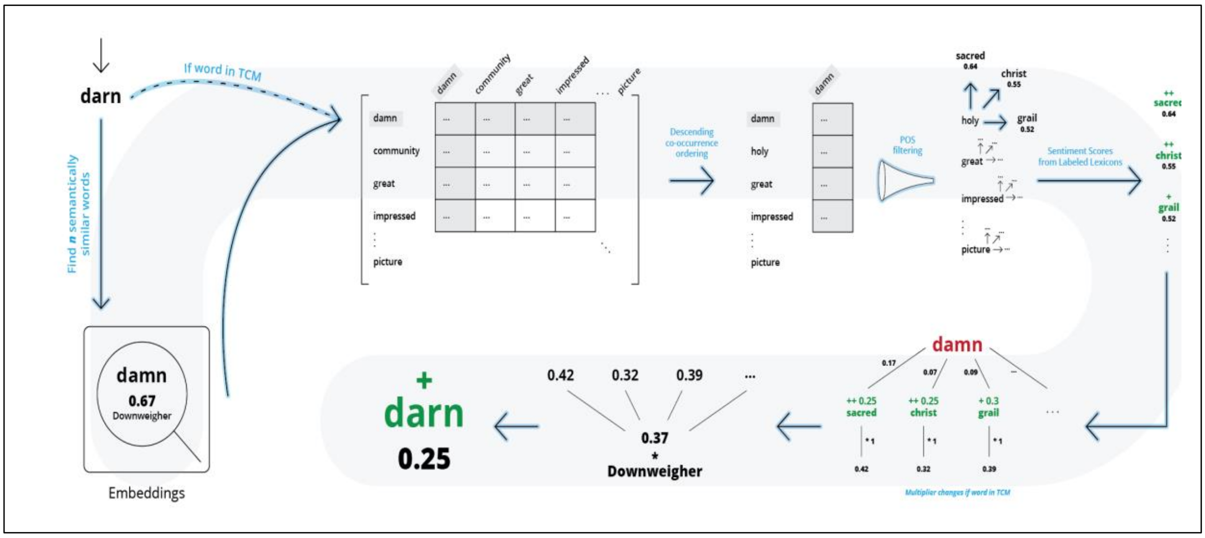

Figure 2: A visual representation of the LexScore algorithm.

2. **Resource-light:** Captures accurate polarity scores with small corpora.
3. **Accuracy Speed Tradeoff:** Allows for manually controlling the tradeoff to address various use-cases.

This work re-scores the Jockers (2015b) and Rinker's augmented Hu & Liu (2004) polarity lexicons for 51 contexts using pre-trained GloVe embeddings (Pennington et al., 2014).

3.1 Downweighing to fit TCM

The first step in this approach is to check if the input word has been used in the corpus. The term-co-occurrence matrix created would be scanned by row and column to find its co-occurrence statistics. This approach introduces a *downweighing* value *D* when the exact word isn't found but a word semantically similar exists in the matrix. The formula used (cosine similarity) is given by

$$D = \cos^{-1}\left(\frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} {A_i}^2}\sqrt{\sum_{i=1}^{n} {B_i}^2}}\right), (2)$$

where A and B are vector representations of the input word and the word most similar to it semantically.

3.2 POS filtering over co-occurrence vector

Once the co-occurrence vectors are extracted in the previous step, the initial domain-specific corpus is checked with the pre-existing lexicon set to find the POS-tags of maximum occurring sentiment bearing words. The co-occurrence vector is then tagged using the R package *tagger* (Rinker, 2015c) that uses openNLP and the Penn Treebank (Marcus et al., 1994). The words whose tags match the previous ones with maximum occurrence are filtered.

Once we have these words with their co-occurrence weights, we now find *n* most semantically similar words from the embeddings and check for their polarity scores in the existing lexicon. We combine these pre-existing polarity scores with their co-occurrence weights and a *multiplier*, which we set at 1.5x of the previous multiplication for the words in the new set matching the previous co-occurrence vector. The final set of the new words that are sentiment-bearing are now semantically compared to the input word.

The formula for calculating the final weights for each word is given by

$$weights = \{(cosine_{sim} + cooccurrence\ weights/100) * multiplier * sentiment\}, (3)$$

where $cosine_{sim}$ is the cosine similarity between the input word and the current word. The last set of words is then averaged (downweighed zero averaging) (Rinker, 2015b) and multiplied with the initial downweighing value D, if it wasn't in the TCM; this is given by the formula

$$\left\{\frac{sum(x)}{sum(x!=0)+\sqrt{\log(1+sum(x=0))}} * D\right\}, (4)$$

which calculates the final sentiment score for the input word in the algorithm.

4 Conclusion

LexScore allows industry and academic researchers to re-score commonly used lexicons to fit their particular domain and context. The paper combines the use of domain-specific co-occurrences with word-embeddings to transform commonly used lexicons to domain-specific ones for opinion mining and research purposes.

The method hopes to help further thoughts in the space of context-dependency of sentiment taking into account the speed and interpretability of algorithms to further be able to scale usage beyond researchers. While the lexicons re-scored by LexScore are not the best, they do provide a more structured way of understanding the sentiment of certain words in different contexts. In the future, this could be coupled with lexicon-inducing algorithms to further improve both scoring and inducing these domain-specific lexicons.

References

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment tree-bank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics.

Guang Qiu, Bing Liu, Jiajun Bu, and Chun Chen. 2009. Expanding Domain Sentiment Lexicon through Double Propagation. In *IJCAI*.

Dirk Hovy. 2015. Demographic factors improve classification performance. In *ACL*.

Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *KDD*. [Huang et al.2014] Sheng Huang, Zhendong Niu, and Chongyang Shi. 2014. Automatic construction of domain-specific sentiment lexicon based on constrained label propagation. *Knowledge-Based Systems*, 56:191–200.

Omer Levy, Yoav Goldberg, and Ido Dagan. 2015. Improving distributional similarity with lessons learned from word embeddings. *Trans. Assoc. Comput. Ling.*, 3.

William L. Hamilton, Kevin Clark, Jure Leskovec, and Dan Jurafsky. 2016a. Inducing domain-specific sentiment lexicons from unlabeled corpora. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics.

Leonid Velikovich, Sasha Blair-Goldensohn, Kerry Hannan, and Ryan McDonald. 2010. The viability of web-derived polarity lexicons. In *NAACL-HLT*.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

A. Esuli and F. Sebastiani. 2006. Sentiwordnet: A publicly available lexical resource for opinion mining. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC'06)*. European Language Resources Association (ELRA).

Tyler Rinker. 2015a. *lexicon: Lexicon Data*. http://github.com/trinker/lexicon

Tyler Rinker. 2015b. *sentimentr: Calculate Text Polarity Sentiment*. http://github.com/trinker/sentimentr

Tyler Rinker. 2015c. *tagger: Part of Speech tagging*. http://github.com/trinker/tagger

Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs up? sentiment classification using machine learning techniques. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP 2002)*.

Linan Qiu. 2016. *Reddit Domain Specific Corpus*. https://github.com/linanqiu/reddit-dataset

Yi Yang and Jacob Eisenstein. 2015. Putting things in context: Community-specific embed- ding projections for sentiment analysis. *CoRR*, abs/1511.06052.

Yi Yang and Jacob Eisenstein. 2017. Overcoming language variation in sentiment analysis with social attention. *Transactions of the Association for Computational Linguistics*, 5:295–307.

Aliaksei Severyn and Alessandro Moschitti. 2015. learning of sentiment lexicons. In *NAACL-HLT*.

Muhammad Zubair Asghar, Au- rangzeb Khan, Shakeel Ahmad, Imran Ali Khan, and Fazal Masud Kundi. 2015. A Unified Frame- work for Creating Domain Dependent Polarity Lex- icons from User

Generated Reviews. *PLOS ONE*, 10(10):e0140204, October.

John A. Bullinaria and Joseph P. Levy. 2012. Extracting semantic representations from word co-occurrence statistics: stop-lists, stemming, and SVD. *Behavior Research Methods*, 44(3):890–907, September.

Sean P. Igo and Ellen Riloff. 2009. Corpus-based semantic lexicon induction with web- based corroboration. In *ACL Workshop on Unsupervised and Minimally Supervised Learning of Lexical Semantics*.

Valentin Jijkoun, Maarten de Rijke, and Wouter Weerkamp. 2010. Generating focused topic-specific sentiment lexicons. In *ACL*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S. Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *NIPS*.

Saif M. Mohammad and Peter D. Turney. 2010. Emotions evoked by common words and phrases: Using Mechanical Turk to create an emotion lexicon. In *NAACL*.

Ethan Fast, Binbin Chen, and Michael S. Bernstein. 2016. Empath: Understanding Topic Signals in Large-Scale Text. In *CHI*.

Lingjia Deng, Janyce Wiebe, and Yoon- jung Choi. 2014. Joint inference and disambiguation of implicit sentiments via implicature constraints. In *COLING*.

Dengyong Zhou, Olivier Bousquet, Thomas Navin Lal, Jason Weston, and Bernhard Scholkopf. 2004. Learning with local and global consistency. In *NIPS*.

Duyu Tang, Furu Wei, Bing Qin, Ming Zhou, and Ting Liu. 2014. Building Large-Scale Twitter-Specific Sentiment Lexicon: A Representation Learning Approach. In *COLING*.

Matthew Jockers. 2015a. Revealing Sentiment and Plot Arcs with the Syuzhet Package. http://www.matthewjockers.net/2015/02/02/syuzhet/

Matthew Jockers. 2015b. *Syuzhet: Extract Sentiment and Plot Arcs from Text*. https://github.com/mjockers/syuzhet

Hu, M., & Liu, B. (2004). Mining and summarizing customer reviews. Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD-2004). Seattle, Washington.

Vasileios Hatzivassiloglou and Kathleen R. McKeown. Predicting the semantic orientation of adjectives. In *EACL*.

J. M. Wiebe, Learning subjective adjectives from corpora, in *Proceedings of the National Conference on Artificial Intelligence, 2000*.

Nicholls and Song. 2009. Improving Sentiment Analysis With Part-Of-Speech Weighting. *Proceedings of the Eighth International Conference on Machine Learning and Cybernetics.*

Peter D. Turney and Michael L. Littman. 2003. Measuring praise and criticism: Inference of semantic orientation from association. *ACM Trans. Inf. Sys.*

Hiroshi Kanayama and Tetsuya Nasukawa. 2006. *Fully Automatic Lexicon Expansion for Domain-oriented Sentiment Analysis.* In Proc. of EMNLP'06.

Ellen Riloff and Jessica Shepherd. 1997. A corpus-based approach for building semantic lexicons. *arXiv preprint cmp-lg/9706013*.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *EMNLP*.

M. Marcus, G. Kim, M. Marcinkiewicz, R. MacIntyre, A. Bies, M. Ferguson, K. Katz, and B. Schasberger. 1994. The Penn treebank: Annotating predicate argument structure. In *ARPA Human Language Technology Workshop*.

Dmitriy Selivanov and Qing Wang. 2018. t*ext2vec: Fast vectorization, topic modeling, distances and GloVe word embeddings in R*. https://text2vec.org