Extracting Sentiment: an Extension of Hamilton et al. (2016)

Jay Kaiser
jckaiser@indiana.edu

Abstract

This work attempted to replicate the corpus-based approaches to constructing strongly domain-specific sentiment-lexicons as outlined in Hamilton et al. (2016). Additionally, the text used for building word-vectors to allow for the construction of such lexicons was preprocessed in several unique ways not found in this original research (i.e., through lemmatization, extraction of part-of-speech tags, etc.). However, the final outputted lexicons obtained by replicating these methods yielded dissimilar lexicons to the original work, most likely due to an unknown bug in some part of the lexicons' construction.

1 Introduction

Sentiment analysis in todays natural language processing suffers from one major issue in particular: though sentiment lexicons lead to astounding accuracies in the domain from which their text is extracted, when exported to an even slightly different domain, accuracies plummet. Therefore, when designing sentiment lexicons, one cannot be solely made for the entirety of the English language; instead, innumerable cross-domain-specific ones must be constructed. The onus is on the researcher therefore to devise a means to automatically derive sentiment lexicons from any domain in an unsupervised manner.

Hamilton et al. (2016) successfully execute such a task, providing a means of acquiring a sentimally-similar lexicon for a specific domain using a set of predefined positive and negative seed words that should maintain their sentiments across all domains. The lexicons created using these methods produce comparable accuracies to state-of-the-art sentiment analysis done through hand-curated sentiment lexicons, but they require only a fraction of the effort involved when doing such. Hamilton et al. utilize the variation found in different Reddit subreddit communities as a proof of concept of their work; these subreddits show such strong variation between one another that cross-domain lexicons would be unable to even attempt an accurate measure of sentiment across them, so domain-specific sentiment lexicons prove the ideal approach in garnering sentiment analysis for them.

However, the work by Hamilton et al. can be extended or optimized in several ways. The researchers mentioned only used raw text in gathering word-vectors to be used for extending the seed set. For the sake of a comparison, research should consider as well text that has been modified by natural language processing technologies (i.e., text that is lemmatized, removed of stop words, marked with part-of-speech tags, etc.). Moreover, research can be done in determining whether the amount or quality of data input into the system has a significant effect the resultant lexicons accuracy. The research done in this paper attempted to verify each of these comparisons among each other, while using the same dataset and methods from the work done by Hamilton et al.

2 Related Work

Sentiment extraction is a valuable field of natural language processing that attempts to extract the polar (i.e., positive or negative) sentiment from raw bodies of text. Applications for such work include tracking shifts in social media and social discussions and inferring consumer-satisfaction in goods and products. Especially in the realms of advertising and intelligence, providing a means to complete fast and accurate sentiment analysis is essential for a deeper understanding of text and its authors. However, as mentioned above, current techniques in extracting sentiment are limited by the domain from which text is extracted, forcing more techniques in the future to rely on obtaining sentiment in more domain-specific scenarios.

A vast body of work has been completed based around extracting sentiment lexicons in unsupervised manners from raw text. Hamilton et al. (2016) use pre-constructed semantic word-embeddings to construct lexical graphs of semantically-similar words; they then apply a random-walk technique derived by Zhou et al. (2004) to propagate sentiment labels across the graphs to build their lexicons. This extends the similar work of Velikovich et al. (2010), who first attempted to propagate sentiment labels across a lexical neighborhood graph, by adding new approaches in graph construction, sentiment label propagation, and word-vector utilization.

Utilizing seed words can result not only in sentiment-lexicons, but any topic-lexicon as well (e.g., violence, government, social media, etc.). Research by Rothe et al. (2016) and Fast et al. (2016) too use similar approaches to Hamilton et al. (2016), utilizing category seed words and corpus word vectors to build topic-specific lexicons. A comparison of each of these techniques could prove valuable when extending this approach for sentiment-extraction in future research.

These methods, among any other methods that use an initial seed-set to induce domain-specific lexicons from unlabeled corpora, are known as corpus-based approaches; they contrast to dictionary-based approaches, which use lexical resources crafted by hand to propagate sentiment from seed labels. Dictionary-based approaches tend toward higher accuracy than corpus-based approaches, given their use of lexical resources made and refined by hand; however, within domains wholly lacking in these resources, only corpus-based approaches are applicable. The most state-of-the-art dictionary-based approach at the time of this research was completed by San Vincente et al. (2014), who too performed label propagation, but over a graph derived from WordNet, a semantic and lexical framework created by Princeton University to graph semantic connections between English nouns and verbs. The work done by Hamilton et al. (2016) outperforms this approach, despite its corpus-based approach.

3 Data

The dataset utilized by Hamilton et al. (2016) is the total number of comments from Reddit in 2014. In this research, a much smaller subset of comments–those from September, 2017–are used instead, both for the interest in the difference size will make as well as for the sake of ease on the part of the researcher. This still amounted to an uncompressed JSON file of 30 GB, which was then converted to a CSV file half that original size. This data was collected automatically by a bot and published freely online.[1]

Reddit[2] is an online community where anonymous users are encouraged to share information on news, interests, and beliefs over certain topics in specified subcommunities (i.e., subreddits) solely dedicated to that topic. Users are then able to comment on posts made by others, as well as mark them and comments as beneficial to the community (i.e., upvote) or mark them as derogatory to the community (i.e., downvote). A final score that marks the sum of the number of upvotes and downvotes for each comment is calculated and placed next to that comment.

These subdivisions between subreddits form natural domains from which this research can be run. Moreover, because each comment is given a score that marks its popularity and relevance in the subreddit, a natural division can be made between popular comments and less popular ones; this division is utilized in the research below.

4 Methods

The original work of Hamilton et al. (2016) provides a starting point for all research done here. Using their descriptions of their algorithms and process, their results are attempted to be replicated. In addition, this research includes comparisons between different types of natural language processing analysis enacted upon the original raw text, (i.e., lemmatization, inclusion of part-of-speech tags, including only popular comments with a score greater than ten, and a combination of tags and popular comments). Originally, it was considered to attempt to build comparative NLP-technique lexicons from text from which all stop words had been removed, but the resulting word-vectors proved too sparse and inaccurate for analysis to be made. All preprocessing has been done through the spaCy Python library.

To convert raw text into sentiment lexicons, a number of techniques are utilized, each of which are explained below. A domain-specific corpus is

[1] https://files.pushshift.io/reddit/comments/

[2] https://www.reddit.com/

first converted into word-embeddings representing the semantic neighborhood from which each token originates. Though the original paper outlines a more original method for discovering these embeddings, in this research Googles Word2Vec is utilized instead. Word2Vec is a two-layer deep neural network that takes a given words context across a corpus and derives appropriate word embeddings for each. Although the authors describe this technique as producing an accuracy statistically-significantly less than their original method, Word2Vec is easily implementable using Python libraries and should prove sufficient for comparison between NLP techniques.

These vectors are then converted into a weighted graph, where each token present in the corpus more than once is represented as a node with edges connecting it to its closest 25 semantic neighbors (based on cosine-similarity). On each of these edges, a weight is placed, as calculated using equation 1 below; this weight utilizes the word-embeddings for each of tokens, as represented by w_i and w_j respectively.

$$E_{i,j} = \arccos\Big(-\frac{w_i^\top w_j}{||w_i||\,||w_j||}\Big) \tag{1}$$

Following the construction of the weighted graph, sentiment labels are propagated across the graph using a random walk method as described by Zhou et al. (2004). With p representing the final sentiment for each token and initialized to 1 divided by the cardinality of the set of unique tokens in the corpus, the formula in equation 2 is then applied iteratively until numerical convergence, where T represents a symmetric transition matrix constructed using the adjacency matrix of the weighted graph created above, and s represents a vector with values that align to members of the seed set set to 1 divided by the cardinality of the seed set and 0 elsewhere. β, which controls the extent to which the algorithm favors local consistency, where similar neighbors are provided with similar labels, is a parameter set here to 0.95 as was done in the original paper.

$$p^{(t+1)} = \beta T p^{(t)} + (1-\beta)s \tag{2}$$

This random walk is done on both the positive and negative seed words, and the final polarity score is given by the positive polarity score divided by the sum of the positive and negative polarity scores. These scores are then normalized across the lexicon to have both zero mean and unit variance.

The seed set is predefined by Hamilton et al. (2016) as domain-neutral positive and negative polarity words. To allow for NLP processing on the original text that include part-of-speech tagging, these seeds had to be separately defined to include their respective tags, all of which are identical to those found in the Penn Treebank, as this is the set of POS tags utilized by spaCy.

5 Evaluation

To evaluate these outputs, the best method would be to run the same semantic word-vectors created using Word2Vec above through the official SentProp lexicon-creator published by Hamilton et al. (2016) alongside their paper, then compare the resulting lexicons and their sentiment values with those extracted here across the various NLP frameworks. Alternatively, given a comparison of only those lexicons created in this project, the impact of each NLP preprocessing technique can be visualized.

Similarly, in their research, Hamilton et al. compare various subreddit's domain-specific lexicons with one another, taking especial interest in rival subreddits whose communities are in direct competition. Upon doing such, the researchers noted stronger similarities between the lexicons of competing subreddits than between those of randomly-chosen subreddits. While it would be interesting to try to replicate these findings, such was not done here, as is explained below.

6 Discussion and Conclusion

The indefinite nature of the section above perhaps foreshadows the main issue that arose upon completing the coding aspect of this research: the results from the analysis done here did not match that of Hamilton et al. (2016). Despite following their methodology and eventually borrowing an extract of their code to finish the project, the resulting lexicons do not scale as did those in their research. The figure below is an example of the most negative and positive five lexicon items for *rTwoXChromosomes*, a subreddit whose users discuss feminism. Note that results similar to these parallel across all ten subreddits and all NLP pre-processing techniques tested here.

...after-abortion-talkline	-9.57
ghost	-9.57
lab	-9.57
from	-9.57
briefly	-9.57
...	...
outdate	0.10
indecent	0.10
footstep	0.10
conductive	0.10
casualconversation	0.10

Note the identical sentiment scores that have been given to seemingly unrelated tokens. Also note the divide between the highest-rated and lowest-rated tokens; the highest go no higher than roughly 0.1, and the lowest go as low as almost -9.6. Not only does this separation happen suddenly in the middle of the lexicon (as opposed to gradually), but this scale vastly differs from those shown by Hamilton et. al (2016) in their own lexicon derived for *rTwoXChromosomes*, shown below.

flu	-4.58
pain	-4.27
cramps	-4.25
nausea	-4.24
cause	-4.23
...	...
sweet	3.44
flair	3.44
wonderful	3.48
beautiful	3.48
lovely	3.69

Naturally, given more time and a fuller understanding of the methodologies outlined in the paper from which this research is based, perhaps it would be possible to discover the specific portion of Hamilton et al.'s method that was miscoded to cause such inaccurate results. Future work perhaps could also utilize the source-code provided in SentProp and uniquely defined word-vectors (e.g., Word2Vec across several forms of NLP preprocessing) to re-emulate the attempts done in this research more successfully.

Though this is an unsatisfying conclusion to the effort provided for this project, much was still learned, both about corpus-based approaches to sentiment analysis and sentiment analysis in general. In addition, all the code (minus one specific portion mentioned above) was rewritten by hand with no regard for the code found in SentProp, so strong practice in Python coding was acquired, despite the unsuccessful outcome.

References

Ethan Fast, Binbin Chen, and Michael S. Bernstein. Empath: Understanding topic signals in large-scale text. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, CHI '16, pages 4647–4657, New York, NY, USA, 2016. ACM.

William L. Hamilton, Kevin Clark, Jure Leskovec, and Dan Jurafsky. Inducing domain-specific sentiment lexicons from unlabeled corpora. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 595–605, Austin, Texas, November 2016. Association for Computational Linguistics.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781, 2013.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'13, pages 3111–3119, USA, 2013. Curran Associates Inc.

Sascha Rothe, Sebastian Ebert, and Hinrich Schütze. Ultradense word embeddings by orthogonal transformation. *arXiv preprint arXiv:1602.07572*, 2016.

Iaki San Vicente, Rodrigo Agerri, and German Rigau. Q-wordnet ppv: Simple, robust and (almost) unsupervised generation of polarity lexicons for multiple languages. 02 2017.

Leonid Velikovich, Sasha Blair-Goldensohn, Kerry Hannan, and Ryan McDonald. The viability of web-derived polarity lexicons. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, HLT '10, pages 777–785, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.

Dengyong Zhou, Olivier Bousquet, Thomas Navin Lal, Jason Weston, and Bernhard Schölkopf. Learning with local and global consistency. In *Proceedings of the 16th International Conference on Neural Information Processing Systems*, NIPS'03, pages 321–328, Cambridge, MA, USA, 2003. MIT Press.