THE SPIN DOCTOR: YOU GIVE US A DOCUMENT, WE'LL REVERSE ITS SENTIMENT

GEORGE FEI, INDIRA PURI, AND CATHY WONG

1. Introduction

It is March 2011, and the Arab Spring is spreading across the Middle East. You are Anna Wintour. Months ago, you commissioned a writer to do an expose on Syria's first lady. This expose was overwhelmingly positive, and its editor has tentatively named the article "A Rose in the Desert." But reports of human rights abuses in Syria have continued to mount, and *Vogue* might receive backlash for running the article. Do you publish the positive expose on Bashar Al-Assad and his wife, or not?

In reality, *Vogue* decided to go ahead with the publication of the article – much to their later regret. The magazine was pummeled for a positive portrayal of a family that increasing evidence showed were brutal, dictatorial rulers. *Vogue* immediately removed the expose, and today, only one copy of the original article is available online (unsurprisingly, the website hosting the article is Gawker).

Vogue could have seriously benefited from software that would take in an article, and, while maintaining its structure and the author's voice, re-written the article to be negative. Creating such software is the goal of our project. For example, given, say a movie review:

"Bambi was a horrible film. Hated it. Shame!"

We would like our software to output the movie review:

"Bambi was a wonderful film. Loved it. Wow!"

The transformation preserves the structure, punctuation, and formatting of the original review – it just makes the review positive. To create this code, we intend to use sentiment analysis methods to classify the sentiment of our new review, thesauruses that have been trained on a large corpus (for example, the Python Natural Language Toolkit (i.e. Stanford's NLTK)) as well as feature extraction methods to determine, for example, how similar the revised review is in structure to the original. Because we write our own evaluation method, we also need to write and implement our own loss function, and stochastic gradient descent function.

To tackle the problem, we use the same corpus of movie reviews used for our second homework, "sentiment." We intend to take each review and reverse its sentiment. Our baseline is simply adding "not" in front of every adjective. Our oracle is a human re-writing the review to preserve the structure, but change sentiment. Our evaluation metric is (did sentiment change) * (closeness in structure) * (reward for using words similar to actual bad reviews). Intuitively, we want sentiment to change; we want closeness in structure; and we want the changed review to look like something a human could plausibly write. To understand the last goal, suppose we have a movie review

"The Matrix was mind-numbing in its awesomeness."

A re-written movie review of the form:

"The Matrix was mind-enhancing in its dullness."

may count as a change in sentiment, and certainly maintains structure, but it doesn't look like anything a human would ever write. Therefore if, as part of our score, we include some evaluation of how similar the words used in our changed review are to actual negative reviews, we discourage outputs like the above.

We defer a precise definition of the evaluation metric to the methodology section. Our baseline is simply adding "not" in front of every adjective and adverb. Our oracle is a human re-writing 20 reviews to switch their sentiment. The baseline receives a startingly low mean evaluation score of 0.038 (median score: 0.357), and the oracle receives evaluation score 0.5187 and median score of 0.742. Because there is a large gap between these, finding a clever way to re-write a review to reverse sentiment is non-trivial.

1.1. Related Literature.

While there is a very large corpus of work on improving sentiment analysis, to our knowledge this is the first paper on reversing the sentiment of a document.

Prior papers have looked at many facets of sentiment analysis, including accuracy [10] [1] [12] [4], domain adaption [2] [9] [13], and how sentiment analysis on various corpii relate to real-world events [3] [11].

Although we focus on sentiment analysis, our work is most closely related to the prior literature on translation. In particular, we may think of our problem as one of translating into a language whose every adjective and adverb has opposite sentiment from normal English. We may therefore utilize statistical machine translation techniques [5], which have been extensively researched. The basic approach behind statistical machine translation techniques is finding a semantic "chunk" of the phrase to be translated, and using Bayes rule with maximum likelihood to find its match in the target language. While the earliest implementation used single-word chunks, later work extended this to bigrams [14], and even more complex semantic structure [8] [6]. Our problem is easier than translation problems in that many of the words between English and sentiment-reversed English are the same; for example, the phrase with opposite sentiment to "the chair" is simply "the chair", because the phrase has neutral sentiment. This means that whereas a traditional translation problem would involve translating every word in a sentence, ours need not do so. We therefore eschew more complicated techniques in favor of easy-to-understand bigram- or single word- reversal.

2. Methodology

2.1. Data.

Our corpus is the Large Movie Dataset, obtained from `http://ai.stanford.edu/~amaas/data/sentiment/` and originally developed by [7]. This dataset contains 50,000 movie reviews. It pre-classifies each movie review on a scale of 1 to 10, where 1 is very negative and 10 is very positive. The dataset also provides pre-processed train and test, so that the movies in each are disjoint. The

(George Fei) Department of Chemistry, Stanford Univeristy.
(Indira Puri) Department of Mathematics, Stanford University. Department of Economics, Stanford University.
(Cathy Wong) Department of Computer Science, Stanford University.
E-mail addresses: georgeqf@stanford.edu, puri@stanford.edu, catwong@stanford.edu.

provided train set is 25,000 reviews, and the provided test set is 25,000 reviews. We use 20% of the pre-processed test set for our dev set. This means that our train set has 25,000 reviews, our dev set 5,000, and our test set 20,000.

2.2. **Tools.** For our evaluation metric, we use a number of off-the-shelf tools: to gauge shift in sentiment, we use the NLTK Vader sentiment classifier to score the transformed documents, and we use the NLTK default tokenizers and part of speech taggers to analyze the document similarities. Future work may also attempt to use predefined NLTK sentiment word dictionaries for transformation, as well as existing SciPy and other model implementations to actually transform the documents.

2.3. **Evaluation Metric.** Our evaluation score is

$$\begin{aligned}\text{Evaluation Score (revised document)} = &\text{ Sentiment Change Score (revised document, original document)}\\ &* \text{Closeness Score (revised document, original document)}\\ &* \text{Uses Typical Language Score (revised document, original document).}\end{aligned}$$

The components of each are listed below.

2.3.1. *Sentiment Change Score.*

$$\text{Sentiment Score Change} = \frac{|\text{Sentiment (revised document) - Sentiment (original document)}|}{\max(\text{Sentiment (original document)} - 1, 10 - \text{Sentiment (original document)})}$$

Intuitively, because our sentiment scores range from 1 to 10, we would like to take any positive review and re-write it to be as negative as possible; and to take any negative review and re-write it to be as positive as possible.

Note that Sentiment Change Score may only take on values between 0 and 1.

2.3.2. *Closeness Score.* We first need to define what is meant by "structure" of a document. When we say that two articles are similar in format and author's voice, we mean that:

(1) The documents contain the same proper nouns.
(2) The documents contain the same number of sentences.
(3) The documents contain the same number of each part of speech (ex. same number of nouns, same number of adjectives, same number of verbs, same number of pronouns).

Therefore a "closeness" score between two documents A and B is

$$\begin{aligned}\text{Closeness}(A, B) = &\text{Proper Noun Score} * \text{Number of Sentences Score}\\ &* \text{Part of Speech Score}\end{aligned}$$

where

(1) Proper noun score $= \frac{|\{\text{proper noun } | \text{ proper noun } \in A \text{ and proper noun } \in B\}|}{\max(\text{number proper nouns in A, number proper nouns in B})}$.
(2) Number of sentences score $= \min\left(\frac{\text{number of sentences in A}}{\text{number of sentences in B}}, \frac{\text{number of sentences in B}}{\text{number of sentences in A}}\right)$.
(3) Part of speech score is

$$\frac{\sum_{\{x_i|\ x_i \text{ is a part of speech ex. noun, adjective, verb,...}\}} |\{b|b \text{ is an } x_i \text{ and } b \in A, B\}|}{\max(\text{Number of words in } A, \text{Number of words in } B)}$$

Note that this means the Closeness Score will always be between 0 and 1.

2.3.3. *Typical Language Score.* In our training corpus, reviews are have sentiment scores 1-10, with 1 denoting very negative, and 10 denoting very positive. If the original review has sentiment 6-10, then define "Relevant Sentiments" to be the range 1-5. If our original review has sentiment 1-5, then define "Relevant Sentiment" to be the range 6-10. Intuitively, because we wish to rewrite the original review to have the opposite sentiment, the relevant sentiments will be those sentiments opposite to the sentiment of the original review.

For each training document t_j, let Bigrams(t_j) denote the set of all bigrams in document t_j. Denote by "Real Bigrams" the set $\cup_{\{t_j|t_j \text{ has a sentiment } \in \text{ Relevant Sentiments}\}}$Bigrams($t_j$). For example, if our original document had positive sentiment (was rated 6-10), Real Bigrams would be the set of all bigrams in negative sentiment (1-5 rated) training set reviews; and if our original document had negative sentiment, Real Bigrams would be the set of all bigrams in positive sentiment training set reviews. Then

$$\text{Typical Language Score} = \frac{\sum_{\text{All bigrams } b_i \text{ in the revised document}} \mathbb{1}\{b_i \in \text{Real Bigrams or } b_i \in \text{Original document}\}}{\text{Number of bigrams } b_i \text{ in the revised document}}.$$

Intuitively, each bigram should be something a real person has written, either in the original review, or in our known set of reviews whose sentiment we are trying to match.

Note that the Typical Language Score will always be between 0 and 1.

3. Results

3.1. **Baseline.** Our baseline algorithm simply transforms the document by adding the word 'not' in front of any adjectives and adverbs identified using the NLTK tagger. This achieves an abysmal mean score of 0.038, but a median score of 0.357. An example can be found in Appendix A.

3.2. **Oracle.** Our oracle measures human performance on this task, using 20 reviews that were handwritten by a team member in an attempt to preserve document similarity while reversing review sentiment. These 20 reviews achieve a mean evaluation score of 0.519 and median of 0.742. An example can be found in Appendix A.

4. Appendix A.

4.1. **Baseline Example.** As stated above, our baseline algorithm attempts to reverse sentiment simply by adding the word 'not' before every identified adjective and adverb in a POS-tagged review.
For example, consider the positive original review:

> I went and saw this movie last night after being coaxed to by a few friends of mine. I'll admit that I was reluctant to see it because from what I knew of Ashton Kutcher he was only able to do comedy. I was wrong. Kutcher played the character of Jake Fischer very well, and Kevin Costner played Ben Randall with such professionalism. The sign of a good movie is that it can toy with our emotions. This one did exactly that. The entire theater (which was sold out) was overcome by laughter during the first half of the movie, and were moved to tears during the second half. While exiting the theater I not only saw many women in tears, but many full grown men as well, trying desperately not to let anyone see them crying. This movie was great, and I suggest that you go see it before you judge.

Using our baseline algorithm, which relies on (an inherently imperfect) default part of speech tagger to identify adjectives and adverbs, we produce the following transformed document, which receives a score of 0.85 by our evaluation metric:

> I went and saw this movie not last night after being coaxed to by a not few friends of mine . I 'll admit that I was not reluctant to see it because from what I knew of Ashton Kutcher he was not only not able to do comedy . I was not wrong . Kutcher played the character of Jake Fischer not very not well , and Kevin Costner played Ben Randall with not such professionalism . The sign of a not good movie is that it can toy with our emotions . This one did not exactly that . The not entire theater (which was sold out) was overcome by laughter during the not first half of the movie , and were moved to tears during the not second half . While exiting the theater I not not not only saw not many women in tears , but not many not full grown men as not well , trying not desperately not not to let anyone see them crying . This movie was not great , and I suggest that you go see it before you judge .

4.2. **Oracle Example.** As our oracle algorithm, we asked a member of our team to manually rewrite 20 reviews, preserving the original document structure as much as possible in line with our defined evaluation metric while still attempting to reverse the sentiment of the document to a highly polarized, opposite sentiment result. This, it should be mentioned, turned out to be a surprisingly difficult task in many cases - there is enormous variation and complexity in how moviegoers express their (often deeply-felt) disappointment and delight, and many of these sentiments have not easy analogue when reversed.
An excellent example of an oracle result that does not, in fact, score well on the current evaluation metric - leaving room for improvement and analysis of our document similarity and other metrics - is the following original and oracle-rewritten review, which received a score of only 0.279:

Original review:

> This is one of the dumbest films, I've ever seen. It rips off nearly ever type of thriller and manages to make a mess of them all.There's not a single good line or character in the whole mess. If there was a plot, it was an afterthought and as far as acting goes, there's nothing good to say so Ill say nothing. I honestly cant understand how this type of nonsense gets produced and actually released, does somebody somewhere not at some stage think, 'Oh my god this really is a load of shite' and call it a day. Its crap like this that has people downloading illegally, the trailer looks like a completely different film, at least if you have download it, you haven't wasted your time or money Don't waste your time, this is painful.

New review:

> This one of the most intelligent films, I've ever seen. It inherits from nearly every thriller and manages to improve a lot on them. Theres not a single bad line or character in the whole piece. If there was a good opportunity, it was an element fully exploited and even in regards to the acting, theres so much good to be said but Ill let to the imagination. I honestly cant fathom the genius that that produces this material and gets it released, does somebody somewhere at some stage realize, Oh my god this is really a trove of gold and calls a producer. Its material like this that has people gasping loudly, the trailers captures completely the atmospheric film, an even if you have downloaded it, you should go to the cinema. Not a waste of time, this is stunning.

References

[1] Albert Bifet and Eibe Frank, *Sentiment Knowledge Discovery in Twitter Streaming Data*, Lecture Notes in Computer Science, 2010, pp. 1–15.
[2] Neil O'Hare, Michael Davy, Adam Bermingham, Paul Ferguson, Paraic Sheridan, Cathal Gurrin, and Alan F. Smeaton, *Topic-dependent sentiment analysis of financial blogs*, TSA '09 Proceedings of the 1st international CIKM workshop on Topic-sentiment analysis for mass opinion (2009), 9–16.
[3] Yu Jiang, Weiyi Meng, and Clement Yu, *Topic Sentiment Change Analysis*, Machine Learning and Data Mining in Pattern Recognition. Lecture Notes in Computer Science **6871** (2011), 443–457.
[4] Alistair Kennedy and Diana Inkpen, *Sentiment Classification of Movie Reviews Using Contextual Valence Shifters*, Computational Intelligence **22** (2006).
[5] P. and Knight Koehn K., *Statistical machine translation* (2009). US Patent 7,624,005.
[6] Yang Liu, Qun Liu, and Shouxun Lin, *Tree-to-string Alignment Template for Statistical Machine Translation*, Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics, 2006, pp. 609–616, DOI 10.3115/1220175.1220252.
[7] Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts, *Learning Word Vectors for Sentiment Analysis*, Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, 2011, pp. 142–150.
[8] Daniel Marcu and William Wong, *A Phrase-based, Joint Probability Model for Statistical Machine Translation*, Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing - Volume 10, 2002, pp. 133–139, DOI 10.3115/1118693.1118711.

[9] Robert P. Schumaker, Yulei Zhang, Chen-Neng Huang, and Hsinchun Chen, *Evaluating sentiment in financial news articles*, Decision Support Systems **53** (2012), 458–464.
[10] Richard Socher, Alex Perelygin, Jean Y. Wu, Jason Chuang, Christopher D. Manning, Andrew Y. Ng, and Christopher Potts, *Recursive Deep Models for Semantic Compositionally Over a Sentiment Treebank*, Conference on Empirical Methods in Natural Language Processing (2013).
[11] Mike Thelwall, Kevan Buckley, and Georgios Paltoglou, *Sentiment in Twitter events*, Journal of the Association for Information Science and Technology **62** (2011), 406–418.
[12] Sida Wang and Christopher D. Manning, *Baselines and bigrams: simple, good sentiment and topic classification*, ACL 2012 Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers **2** (2012), 90–94.
[13] Rui Xia, Chengqing Zong, Xuelei Hu, and Erik Cambria, *Feature Ensemble Plus Sample Selection: Domain Adaption for Sentiment Classification*, IEEE Intelligent Systems **28** (2013), 10–18.
[14] Richard Zens, Franz Josef Och, and Hermann Ney, *Phrase-Based Statistical Machine Learning*, Advances in Artificial Intelligence. Lecture Notes in Computer Science **2479** (2002), 18–32.