Extracting Sentiment: A Replication of Hamilton et al., (2016)

Jay Kaiser

Hamilton et al., (2016)

- Sentiment-analysis is domain-specific, and attempting cross-domain analysis yields poor results.
- Domain-specific lexicons can be made by hand, but this is timely and expensive.
- Therefore, we need a way to create domain-specific lexicons in an unsupervised manner.

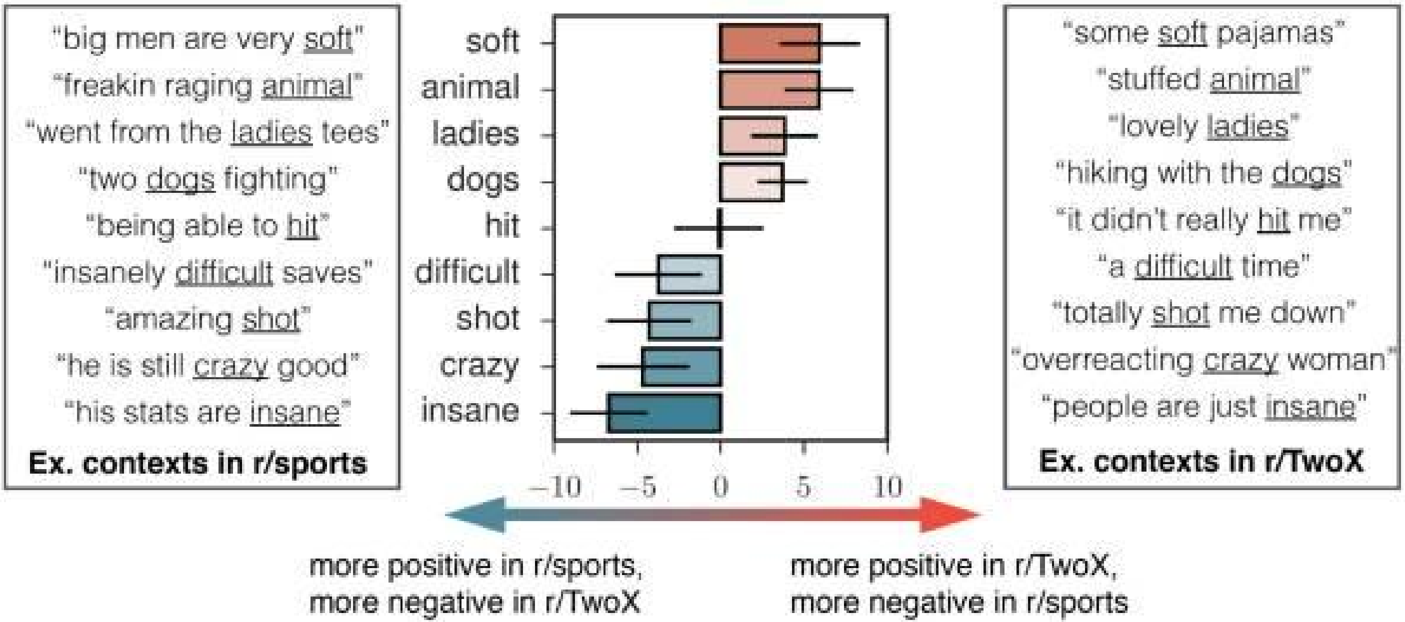
Hamilton et al., (2016)

Seed words

The seed words were manually selected to be context insensitive (without knowledge of the test lexicons).

Domain	Positive seed words	Negative seed words
Standard English	good, lovely, excellent, fortunate, pleasant, delightful, perfect, loved, love, happy	bad, horrible, poor, unfortunate, unpleasant, disgusting, evil, hated, hate, unhappy
Finance	successful, excellent, profit, beneficial, improving, improved, success, gains, positive	negligent, loss, volatile, wrong, losses, damages, bad, litigation, failure, down, negative
Twitter	love, loved, loves, awesome, nice, amazing, best, fantastic, correct, happy	hate, hated, hates, terrible, nasty, awful, worst, horrible, wrong, sad

Hamilton et al., (2016)



Dataset

Reddit comments from a number of subreddits from 09-2017.

Will compare the following:

- Raw text minus stopwords
- “Popular” comment text minus stopwords ($n > 10$)
- Lemma-POS tag combos minus stopwords
- “Popular” lemma-POS tag combos minus stopwords ($n > 10$)

Dataset Conversion

Raw	“But Watergate destroyed his ability to pass legislation.” “But a reasonable conversation needs to be had about how we can reduce gun related crime and death…”
-Stops	“Watergate destroyed ability pass legislation.” “Reasonable conversation needs reduce gun related crime death…”
+Popular	“Watergate destroyed ability pass legislation.”
+Lemma, POS	“watergate-NNP destroy-VBD ability-NN pass-VB legislation-NN”

Hamilton’s Vector Space Model

“The first step [in our approach is] **building** [high-quality semantic representations for]...”

$$\mathbf{M}_{i,j}^{PPMI} = \max\left\{\log\left(\frac{\hat{p}(w_i, w_j)}{\hat{p}(w)\hat{p}(w_j)}\right), 0\right\}$$

“p^ is the smoothed empirical probability of word co-occurrences within the window of text.”

The truncated singular value decomposition of $\mathbf{M}^{\mathbf{PPMI}}$ is found, with vector embeddings of dimension 300.

Each word $\mathbf{w}_i^{\mathbf{SVD}}$ is found by $(\mathbf{U})_i$.

My Vector Space Model

That looks terrible, and I have other projects to work on.

I’ll just use Word2Vec.

Though their method outperforms Word2Vec, I’m not trying to publish this.

Weighted Graph

A weighted graph is formed given each seed word and its nearest 25 semantic neighbors, using cosine-similarity.

Sentiment labels are propagated from this graph using a random-walk method.

"A word's polarity score is proportional to the probability of a random walk from the seed set hitting that word."

Weighted Graph

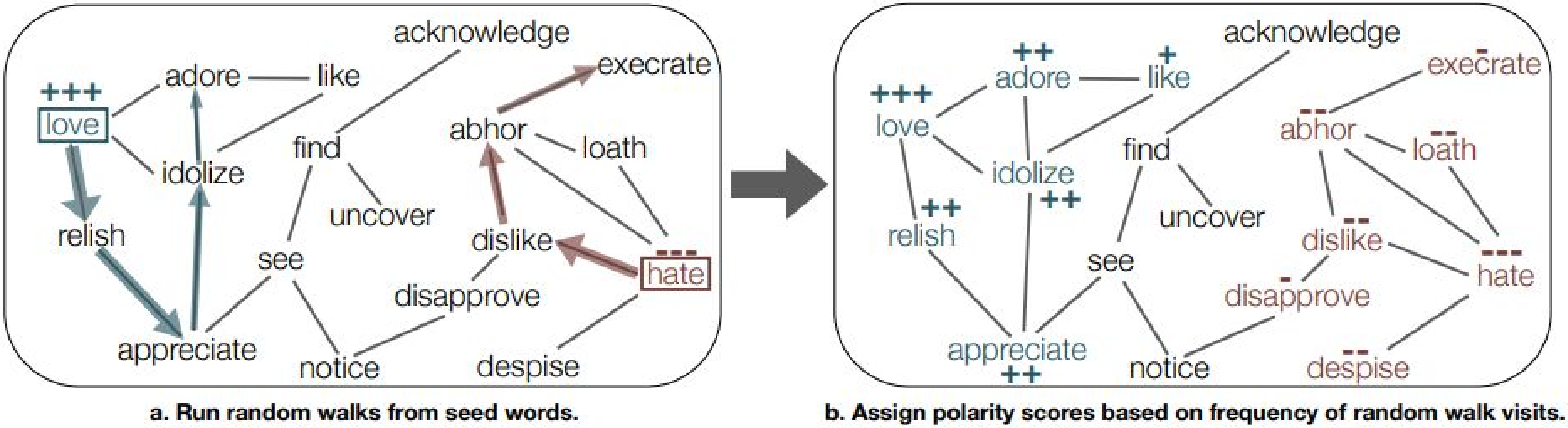


Figure 3: Visual summary of the SENTPROP algorithm.

Outcome

- I will compare the results of each of the text-types with each other, and with the official SentProp created from this paper.
- Hamilton et al. compare sentiments from conflicting subreddits and find greater similarity in their lexicons than in unrelated subreddits.

r/democrats	r/Republican
r/hillaryclinton	r/The_Donald
r/TwoXChromosomes	r/TheRedPill
r/mylittlepony	r/sports

Summary

- I am replicating the results from the first part of Hamilton et al., (2016).
  - No lexicon-duplication, temporal comparison, etc.
- I am including further NLP processing to the text (lemmatization and POS tagging).
- I am comparing popular comments versus all comments.

Questions?