Learning Polarity Embedding

Zhiping Xiao [*]
ScAi Lab
CS Department, UCLA
patricia.xiao@cs.ucla.edu

Zhicheng Ren
ScAi Lab
CS Department, UCLA
renzhicheng1996@gmail.com

Zijie Huang
ScAi Lab
CS Department, UCLA
zijiehuang@cs.ucla.edu

Abstract

Our goal is to extract the political-polarity information from the politicians' speech, which could make supporters of different parties use totally different expressions referring to the same concept. We formalize this problem as learning the polarity embedding of the words. Uulike the standard methods where all aspects of a word's meaning composites the word vector representation, we want it to specifically capture two parts: (1) the political embedding, and (2) the non-political, semantic embedding. In theory we can use any existing model as our base model. For instance, word2vec, GloVe, ELMo, BERT, XLNet. Based on that, we propose to add auxiliary tasks to specifically force the embeddings to end up in what we want. At the current stage, we have the previously-crawled tweets, the most-recent $3,000$ tweets for the 115^{th} and 116^{th} representatives before March, 2019. Our experiments show that there still remain some challenges to conquer, and also show that the existing approaches are not good enough in this scenario.

Our code and current-stage data are released at https://github.com/PatriciaXiao/CS263_PolarityWordEmbedding.

1 Introduction

We aim at learning separate embeddings of one word for its political polarity and its political-free semantic meaning.

The idea of this project came from some observations of an existing dataset we crawled from Twitter in 2019. Previously, the text part of that dataset was not fully utilized. There could be a lot of interesting topics to be covered by analyzing the text information we have, among which we found it interesting to work on word embedding.

And it comes naturally that it'll be a waste if we just do ordinary embeddings. As is shown in Table 1, we saw that politicians from the democratic party and the politicians from the republican party have some very different habits of expression. We therefore propose to learn an embedding, where part of it captures the political-neutral word-semantic embedding, while the other part captures the political information.

The main challenge we are to face is the lack of models solving this kind of problems, especially considering the specific social-media polarity embedding setting. On the other hand, the novelty by itself will make our project interesting.

Inspired by GN-GloVe (Zhao et al., 2018), which is based on the GloVe (Pennington et al., 2014) model, we re-implemented it in PyTorch, applying it directly to our political datasets. Then we examined the results, found some shortcomings of the model under this scenario, and tried making some improvements accordingly. We also proposed a future direction to explore, together with a new perspective of interpreting the GN-GloVe model.

2 Related Works

2.1 Self-Supervised Language Model

We call the base models of learning word embeddings the self-supervised language models. The reason is obvious: there's no additional label to any text information in the very basic word embedding examples. There, co-occurrence information is the most important information from the text to train itself on the word-level embedding tasks. It is true for all possible base models we might use, from word2vec, GloVe, ELMo, to BERT and XLNet.

We are probably using only word2vec and GloVe, due to the limited computational resources at this period.

[*]The author names are in alphabetic order, except for the first name being the team leader name.

2.2 GN-GloVe

GN-GloVe ((Zhao et al., 2018)) is one of the few word embedding models that has similar idea as ours. It is considered one of the most important related works to our problem, and could potentially be where we start exploring from.

In their setting, the authors tried to capture the gender-related embedding, and separate it from gender-neutral embedding.

Just as the name suggests, the model is based on GloVe ((Pennington et al., 2014)). It is not a fine-tuning model on the pre-trained GloVe embeddings, rather, it is a pre-training model. There are two auxiliary losses added to the standard GloVe loss, one of them contributes to diverging the gender dimensions, one of them tries to keep the semantic part of the embedding in the null space of the gender embedding.

$$J = J_G + \lambda_D J_D + \lambda_E J_E$$

The loss contains three parts:

1. J_G, the standard GloVe loss

   In the paper the loss is:

   $$J_G = \sum_{i,j=1}^{V} f(X_{i,j})\big(w_i^T \tilde{w}_j + b_i + \tilde{b}_j - \log(X_{i,j})\big)^2$$

   where V is the vocabulary size, $X_{i,j}$ is the values in the concurrence matrix. And the weighting term $f(X_{ij})$ is defined as:

   $$f(x) = \begin{cases} (x/x_{max})^{\alpha} & \text{if } x < x_{max} \\ 1 & \text{otherwise} \end{cases}$$

   where x_{max} and α are hyper-parameters.

2. Loss that maximizes the difference of the polarity dimension (J_D)

   There are two options for J_D presented in GN-GloVe paper:

   $$J_D^{L_1} = -\Big\| \sum_{w \in \Omega_{female}} w^{(g)} - \sum_{w \in \Omega_{male}} w^{(g)} \Big\|_1$$

   $$J_D^{L_2} = \sum_{w \in \Omega_{female}} \|\beta_1 \mathbf{e} - w^{(g)}\|_2^2 + \sum_{w \in \Omega_{male}} \|\beta_2 \mathbf{e} - w^{(g)}\|_2^2$$

   where $\mathbf{e} \in \mathbb{R}^k$ is a all-one vector, $\beta_1 = -1$, $\beta_2 = 1$ (the reverse of the original paper, for user's habit). Both were tested, but there weren't a concrete conclusion on which one is the best. We decided to test on both.

3. Loss that keep the neutral embedding neutral (J_E)

   In this part, they tried to align the non-polarity dimensions $w^{(n)}$, or we can call it the neutral/semantic parts, to be polarity-free. J_E encourages $w^{(n)}$ of the words to stay in the null-space of the polarity direction, where the polarity direction is measured by the political pairs' $w^{(n)}$ differences.

   $$J_E = \sum_{w \in \Omega_{neutral}} (v_g^T w^{(n)})^2$$

   where v_g is calculated **per epoch** as:

   $$v_g = \frac{1}{|\Omega'|} \sum_{(w_{female}, w_{male}) \in \Omega'} (w^{(n)}_{female} - w^{(n)}_{male})$$

However, there are some challenges we need to overcome as we adapt this idea to our own problem. The most obvious challenge is that, there isn't very good political-polarity pairs. One can easily use GN-GloVe to find gender-polarity pairs such as "waiter-waitress", and latter on conclude a beautiful conclusion such as: "doctor" seems to infer male and "nurse" seems to infer female. But how could we find political-polarity word pairs like such remains a problem to be considered. We argue that, at least we must consider using phrases.

2.3 Phrase-Parsing

In order to utilize phrases, we used AutoPhrase ((Shang et al., 2018)) for phrase-parsing. Unlike previously-proposed methods, where human-experts' involving is required, AutoPhrase is able to automatically mine the phrased from a given piece of text.

They did phrase-mining by proposing potential phrases from general knowledge base, and estimating the potential phrases according to the following criteria:

- Popularity: high frequency in the given document collection.
- Concordance: probability occurring together significantly higher than expectation.
- Informativeness: indicative of a specific topic or concept.

- Completeness: Long frequent phrases and their subsequences within those phrases may both satisfy the 3 criteria above.

Beyond utilizing the general knowledge base text, they also applied a part-of-speech (POS) tagger to help enhance the prediction quality. The official code of AutoPhrase is released on Github at https://github.com/shangjingbo1226/AutoPhrase.

3 Problem Definition

Let's say that we have a word w, represented in the d-dimensional vector space by $v \in \mathbb{R}^d$.

We want v to be represented by:

$$v = \left[v^{(n)}; v^{(p)}\right] \in \mathbb{R}^{d_n} \times \mathbb{R}^{d_p}$$

where $d_n + d_p = d$, $v^{(n)}$ represents the neutral embedding, $v^{(p)}$ represents the political embedding.

Our goal is to learn the embedding in this special structure, from Twitter dataset.

4 Method

4.1 Relaxed GN-GloVe Model

Based on GN-GloVe's shortcomings in political-polarity embedding-learning, we introduced our own method.

First, we take an element-wise average over all the loss components.

After this adjustment, the GloVe loss becomes:

$$J_G = \frac{1}{|V|} \sum_{i,j=1}^{V} f(X_{i,j}) \left(w_i^T \tilde{w}_j + b_i + \tilde{b}_j - \log(X_{i,j})\right)^2$$

In our case, because of the difficulty of finding proper pairs, we decided to relax the setting of using pair of words into using sets of words. Which means that, we will use a set of democratic words, and another set of republican words, to compute the polarities. Therefore, J_D becomes:

$$J_D^{L_1} = -\left\| \frac{1}{|\Omega_{democratic}|} \sum_{w \in \Omega_{democratic}} w^{(g)} - \frac{1}{|\Omega_{republican}|} \sum_{w \in \Omega_{republican}} w^{(g)} \right\|_1$$

$$J_D^{L_2} = \frac{1}{|\Omega_{democratic}|} \sum_{w \in \Omega_{democratic}} \|\beta_1 \mathbf{e} - w^{(g)}\|_2^2 + \frac{1}{|\Omega_{republican}|} \sum_{w \in \Omega_{republican}} \|\beta_2 \mathbf{e} - w^{(g)}\|_2^2$$

For J_E, we take the average instead of the sum again, and calculate v_g every INT (a constant) number of **batches**:

$$J_E = \frac{1}{|\Omega_{Neutral}|} \sum_{w \in \Omega_{Neutral}} (v_g^T w^{(n)})^2$$

batches instead of epochs are used, because of the nature of our dataset. Our dataset is not such big, and do not require many epochs of training. Having too large an interval could also cause bad performance.

4.2 Future Exploration

From our perspective, we notice that GN-GloVe could be regarded as a multi-task learning framework of solving the polarity (in their case, gender-polarity) word embedding problem.

We regard their model as having three tasks:

1. Task #1 of self-supervised learning according to the co-occurrence of words ($\mathcal{L}_G$, the standard GloVe loss)

2. Task #2 that maximizes the difference of the polarity dimension ($\mathcal{L}_D$)

3. Task #3 that keep the neutral embedding neutral ($\mathcal{L}_E$)

From this perspective, we might change everything of GN-GloVe model, including the base model – it doesn't has to be GloVe anymore. And we will change the loss as well, so that it is probably another form of joint-learning, or being more than just joint-learning. But we keep the idea and the framework of having at least three tasks learning jointly.

There's some uncertainty on whether or not the Twitter dataset, or, we can name it "political casual talk" dataset, is too noisy for some models. We might also consider including more formal text data to help learn the embedding, such as the congressman speech. There must be a lot of those resources.

5 Experiments

5.1 Dataset

The name list of the politicians and their profile information are collected from the official website at https://www.congress.gov/members. We added Obama's and Trump's cabinets in, together with the presidents themselves. Then we

party	democratic	republican
vocabulary	336962	338188
tokens	15546896	13938448
# unique hashtags	66171	59555
# hashtags	481232	443768
# max hashtags per tweet	40	17
# unique emoji	1320	1307
# emoji	22362	27710
# max emoji per tweet	140	78
retweets ratio	28.00%	27.16%
$\#retweet + \#quote$	563045	573837
$\frac{\#retweet}{\#retweet+\#quote}$	13.24%	8.85%
$\frac{\#retweet+\#quote}{\#tweets}$	47.29%	32.59%

Table 1: Analyzing the politicians' tweets dataset we have. *Quote* refers to the retweeting behavior with some comments, while *retweet* itself is retweeting without comment.

find the approximately 600 politicians' Twitter accounts, turned out that only 585 of them have one, among which 2 are *Independent* politicians, the rest belong to either *Democratic* or *Republican* party.

We crawled the most recent $\approx 3,200$ tweets of each of the politicians, split the tweet contents into two parties that the one who made the post belongs to. Then we found some interesting features that seemingly tells us something, concluded in Table 1. As is indicated there, it seems that democratic politicians generally love hashtags more, republican politicians prefer to put comment when retweeting (called a "*quote*" in this case) more than democratic politicians.

5.2 Environment

We are planning to use PyTorch, of course with Python (version 3.7). All team members are from the ScAi lab, so we have a few servers that could potentially be used for this course project. Our server has CUDA 10.0 installed, and NVIDIA Titan Xp GPUs with $11G$ memory each.

5.3 Phrase-Parsing Results

AutoPhrase did pretty well in our case. The top few phrases that it came about are as listed in Table 2. Obviously, most recognized phrases are names. In our case it will be pretty useful in theory, for that we do not want to split a user's name into separate tokens. Besides, other kinds of results also make a lot sense. Such as "deepwater horizon" ranked 38^{th}.

Confidence Score	Phrase
0.9901403703	roy moore
0.9895646105	golden triangle
0.9893682485	abraham lincoln
0.9893044339	christopher steele
0.9892777179	martha mcsally
0.9890479972	kay hagan
0.9889986674	luther strange
0.9889975703	niagara falls
0.9889752848	justin amash
0.9889752848	linda mcmahon

Table 2: The top-10 mined phrases

At this stage we decide that using these results are beneficial.

5.4 GN-GloVe Model for Political Polarity

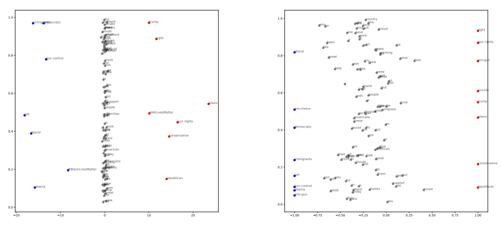

Figure 1: Visual illustration of the embeddings of some selected words. Left: political-polarity-dimension's embedding; Right: political-polarity-dimension's embeddings. The y-axis are just random variables. Colors represents the parties, blue for democratic and red for republican.

Following the GN-GloVe paper, we re-implemented their code in PyTorch, and used ℓ_1-version of J_D by default, since from their paper, it looked like the $J_D^{\ell_1}$ works better than $J_D^{\ell_2}$.

We found the following pairs of words as listed in Table 3:

We use these pairs as the anchors, set the total dimension of the word embeddings to be $d = 200 = d_n + d_p$, among which the polarity dimension is set to be $d_p = 1$, the remaining $d_n = 199$ dimensions are expected to be neutral. The results of running GN-GloVe is as shown in Figure 1.

We found that this design makes the words' embeddings too extreme. At first we thought that it could be because of the use of $J_D^{\ell_1}$ loss component, but latter on we found that switching to $J_D^{\ell_2}$ does not help that much. It was simply, instead of dragging the polarity dimension of the anchor

D words	R words
Democratic	Republican
Obama	Trump
Right	Left
Liberal	Conservative
Pro-LGBT	Anti-LGBT
Pro-Choice	Pro-Life
Immigrants	Aliens
Gun control	Gun rights
Universal healthcare	Private healthcare
#BlackLivesMatter	#AllLivesMatter

Table 3: The political word pairs. Because of the nature of Twitter, we treat hashtags as word tokens as well.

words to $\pm 10 \sim 20$, it force them to be around ± 1, but the remaining words do not show any useful information in their polarity dimension.

We then realize that it was because of our word pairs are of very low quality, compared with the gender-related dataset, which we already discussed in Section 2.2. This finding motivated us into the next iteration — a relaxed model.

5.5 Relaxed GN-GloVe Model for Political Polarity

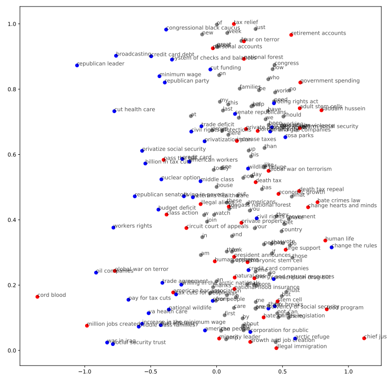

Figure 2: Visual illustration of the political-polarity embeddings of the selected words.

Since GN-GloVe relies too heavily on the word-pairs' quality, we consider changing the design of their model, relaxing it into using word sets. Details are discussed in Section 4.1.

Because of that the sets are not guaranteed to have the same amount of elements inside, we decided that we have to take an average. After taking the average, though, it revealed a shortcoming of the $J_D^{\ell_1}$ design: it could potentially grow into **negative** loss. We know that it doesn't make sense to have negative looses anyhow, but the design of $J_D^{\ell_1}$, as we can see, makes this component negative.

Previously, if we do not take an average over the words, then $J_D^{\ell_1}$ will be much smaller than the sum of J_G and $\lambda_E J_E$. But now, after averaging out, it could be dominant.

Therefore, we switched to the design of $J_D^{\ell_2}$.

All other hyper-parameters and settings were kept the same as before.

The results, as is shown in Figure 2, doesn't look significantly more reasonable. However, the polarity scores are not that extreme, which is probably a good signal. On the other hand, even if we consider the words as sets, the quality of such polarity measurement remains doubtful, and could potentially confuse our model. We still need more explorations on the embeddings to tell whether or not this model performs well on our task as we expected.

5.6 Sentence-Classification Task

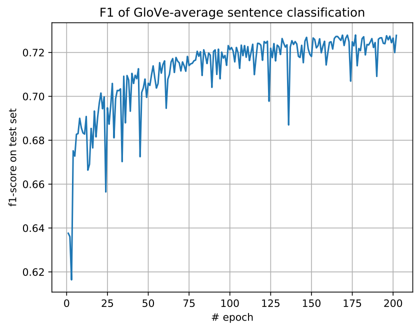

Figure 3: Visual illustration of the F1-score on testing dataset to do simple sentence-level classification task.

As a downstream task, we would like to examine the embeddings' quality based on how well those word-level embeddings are in telling a tweet's political tendency.

Quite intuitively, we came up with the sentence-level classification task to do so.

We use the learned embedding, together with a simple 2-layer MLP structure, to conduct sentence-level classification task. The input representations of the sentences are calculated as simple as an average of all the content words' embeddings.

The performance is shown in Figure 3. As you can see, the performance is actually not much better than using random embeddings as inputs of the MLP classifier, where we can expect to see the binary-classification results of some random vectors to be around 50% accuracy.

This sentence-level result simply shows that the embedding is better than random, but not yet good enough to tell that it properly captures the polarity information.

6 Conclusion

From the early exploration, we found the GN-GloVe model does not automatically fit into the scenario of political-polarity embedding, which, in a way, could be regarded as a way-more-challenging task to do, that reveals the blind area of the existing models today. On the other hand, GN-GloVe shed light on a possible direction to go for solution on it. We will keep on exploring the multi-task design. We are believe that we are on the right track towards some promising results.

References

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Jingbo Shang, Jialu Liu, Meng Jiang, Xiang Ren, Clare R Voss, and Jiawei Han. 2018. Automated phrase mining from massive text corpora. *IEEE Transactions on Knowledge and Data Engineering*, 30(10):1825–1837.

Jieyu Zhao, Yichao Zhou, Zeyu Li, Wei Wang, and Kai-Wei Chang. 2018. Learning gender-neutral word embeddings. *arXiv preprint arXiv:1809.01496*.