Surveying Public Opinion using Label Prediction on Social Media Data

ASONAM, Vancouver, August, 2019

Marija Stanojevic, Jumanah Alshehri, Zoran Obradovic

Temple University, Philadelphia, PA, USA

Presented by: Marija Stanojevic

August 29, 2019

Introduction (1)

- **Motivation**: short-text classification models typically require large labeled data making proper social media mining an impossible task for most of the academic and industry organizations (due to resources required for labeling).
- If this challenge is solved, we can use social media data rather than or together with surveys to better understand public opinion, leading to robust decisions on polices and products.
- Social media texts (including product reviews) are typically very short, include typos, cynicism, jargon and emoticons.
- They often contain or refer to pictures, videos, posts or people.

Introduction (2)

- We can use clustering to understand unlabeled texts by dividing it into groups, however, it is not guarantied that clusters would be meaningful to our topic of interests.
- Instead, we suggest semi-supervised framework that uses a little amount of labeled data to predict labels for the other texts.
- **Objective 1**: Classify social media posts into groups (opinions) given a small seed (labeled examples).
- **Objective 2**: Utilize existing supervised methods and their superior performance.
- **Objective 3**: Make a framework that can utilize different supervised methods.

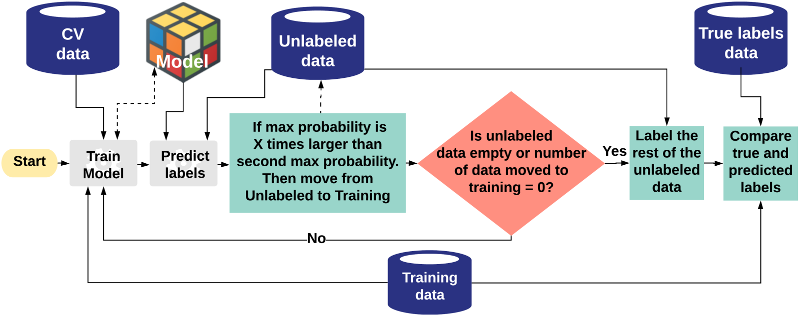
TEMPLE UNIVERSITY®

Proposed Framework

- Classification is easy for some examples, but very hard (even for humans) for the others
- Would you classify this as positive or negative: **"The tacos shells were mainly broken. shells are too thin for transportation. the shells are very good. mainly use for dipping."**
- What would be your recommendation score based on this text?
- **Idea**: Let's do the easy job first and add those to the training corpus. Then, we try to classify harder examples, given more labeled data that we have.

TEMPLE UNIVERSITY®

Proposed Framework



Architecture of the proposed semi-supervised label prediction (SLP) framework.

Proposed Framework

- **Training model** can be any model that can predict labels. In our work, we choose three distinct options:
  - Logistic regression of tf-idf features of n-grams (tfidf+ngrams) [1]
  - fastText (supervised version of word2vec, skip-gram) [2]
  - VDCNN (very deep convolutional neural network) [3]
- Those methods have the best performance for certain size of data and they are ordered by the amount of data they require to work well and by success they have given required amount of data is feed to them.

[1] X. Zhang, J. Zhao, and Y. LeCun, "Character-level convolutional networks for text classification," in Ad- vances in neural information processing systems, 2015, pp. 649–657.

[2] T. Mikolov, E. Grave, P. Bojanowski, C. Puhrsch, and A. Joulin, "Advances in pre-training distributed word representations," in Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018), 2018.

[3] A. Conneau, H. Schwenk, L. Barrault, and Y. Lecun, "Very deep convolutional networks for text classification," arXiv preprint arXiv:1606.01781, 2016.

Proposed Framework

- By adding labeled examples to the training set in the future iteration, we are always adding a point that was easy to classify.
- If only search space is considered, we didn't add new information and algorithm shouldn't perform better
- However, adding new data adds new contexts and relations between words that are beneficial for the models that are able to work with contexts and co-occurrences.
- We expect tfidf + ngram algorithm to always get worse results after our framework is applied due to increased noise
- However, we expect that fastText and VDCNN get better, despite the noise and thanks to the new contexts we introduced.

Data

- We use four standard datasets, which are balanced, labels for all examples are known.
- To mimic social-science researcher, we download guns related data from twitter based on certain key words with a goal to understand public opinion on guns control/rights.

Table: Large-scale text classification datasets

Dataset	Unlabeled	Data Type	Classes
AG news	128k	Topics	4
Yelp Review Polarity	598k	Reviews	2
Amazon Review Full	3,650k	Sentiment	5
Amazon Review Polarity	4,000k	Sentiment	2
Twitter guns	11,750k	Opinions	2

Twitter Data and Proposed Seed Labeling

- Tweets are selected using expert given hashtags.
- As opposed to perfect public, dataset is heavily unbalanced (92:8 for guns rights) due to experts knowledge bias and posting activity of the two sides.
- We got much more tweets (11,750 K), but no labels.
- To speed up seed labeling, we find profiles which posts are only advocating one of the opinions (88 for gun rights and 83 for gun control) and label their posts accordingly (7,782 for guns control and 666 for guns rights).

TEMPLE UNIVERSITY®

Experimental Settings

- **Ratio** is a threshold that determines if example will be accepted as training data in next round. It is a ratio of the label probability of two most probable labels. Tested ratios are: 1.2, 1.5, 2, 3 and 4.
- **Seed size**: 500, 5,000 or 50,000 labeled examples to start with.
- **Exit criteria**: It finishes if there are no more unlabeled examples or if no data is moved to training set in the last iteration.
- To speed up algorithm, we can exit if size of unlabeled dataset is $< X$ or if number of examples moved to training set is $< Y$.
- This is benefitial for deep learning algorithms (batch size).
- All examples left unlabeled are labeled with the most probable label in the last iteration.

Results (1)

Model	AG	Amz. F.	Amz P.	Yelp P.	Twiter
VDCNN	0.375	0.286	0.683	0.790	0.921
SLP(VDCNN)	0.645 (4)	0.285 (2)	0.758 (3)	0.521 (4)	**0.965 (4)**
ngrams+tfidf	0.729	0.270	0.744	0.776	0.954
SLP(ngrams+tfidf)	0.721 (1.2)	0.311 (1.2)	0.731 (1.2)	0.807 (1.2)	0.927 (1.2)
fastText	0.336	0.201	0.500	0.500	0.920
SLP(fastText)	**0.855 (2)**	**0.409 (2)**	**0.817 (2)**	**0.810 (2)**	0.923 (4)
SLP(fT+ngrams+tfidf)	0.797 (3)	0.354 (1.2)	0.771 (3)	0.744 (1.2)	0.951 (1.5)

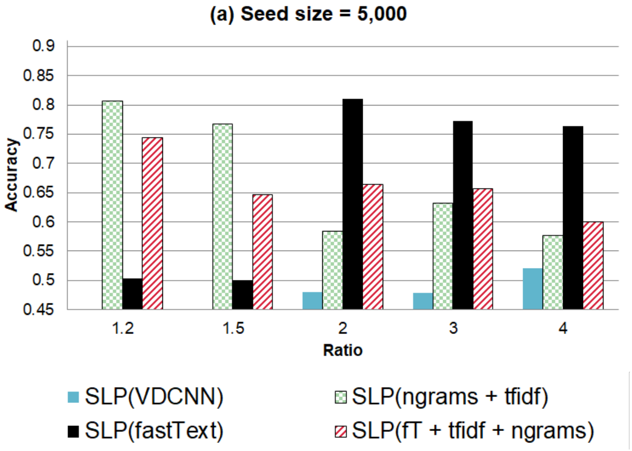
Label prediction accuracy (LP) for seed size = 5,000. For each value of the proposed model, the best ratio for which this value was obtained is given in brackets.

Results (2)

Model	**AG**	**Amz. F.**	**Amz P.**	**Yelp P.**
VDCNN	0.869	0.458	0.862	**0.908**
ngrams+tfidf	0.757	0.388	0.749	0.805
fastText	0.853	0.387	0.809	0.501
SLP(VDCNN)	**0.884(3)**	**0.482(2)**	**0.876 (4)**	**0.908(3)**
SLP(ngrams+tfidf)	0.750(4)	0.372(4)	0.731(1.2)	0.809(4)
SLP(fastText)	0.871(2)	0.431(2)	0.840 (2)	0.863(2)
SLP(fT+ngrams+tfidf)	0.861(4)	0.398(1.2)	0.816(4)	0.808(1.2)

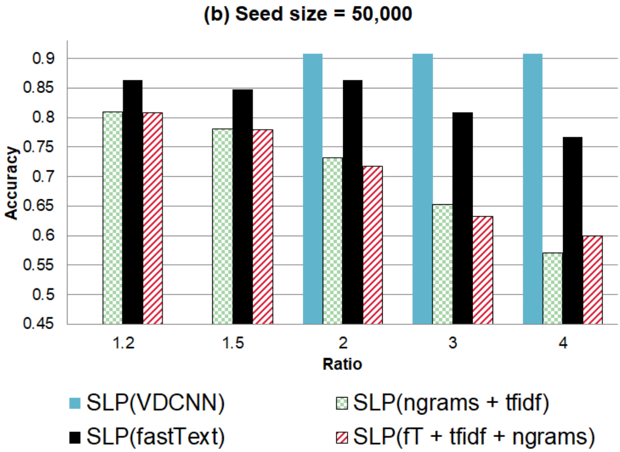
Label prediction accuracy (LP) for seed size = 50,000. For each value of proposed model the best ratio for which this value was achieved is given in brackets.

Results (3)



Accuracy of SLP model trained with multiple language modeling algorithms and different ratios for seed size = 5,000 (Yelp P.).

Results (4)



Accuracy of SLP model trained with multiple language modeling algorithms and different ratios for seed size = 50,000 (Yelp P.).

Results (5)

- Unfortunately, model doesn't perform well for seed size of 500 because there aren't enough examples to learn the useful characteristics for classification, especially with short texts. If it is not possible to label more data, ngrams + tfidf should be used.
- F1 score is measured on Twitter data because of imbalance. Best result (0.675) is achieved with SLP(ft+ngrams+tfidf).
- SLP with fastText performs the worst from all SLP models, which is expected since fastText is not able to handle imbalance.
- Additionally, we tested our models on DBPedia dataset and models were not able to classify well due to huge number of classes (14) and a little common useful information between the examples.

Results (6)

- In the last year many novel models are proposed for language understanding that can also be used for classification (BERT, ULMFiT, ELMo). Those models can also be incorporated into our framework and can benefit from it.
- Our model is helpful even for huge amount of data, however improvement it brings is small (up to 1%).
- In many cases of opinion modeling, we are not interested in classifying hard examples because those are mixed opinions. Our framework allows to discard such examples and in that case achieves 3-5% better accuracy on the data that is classified.

Summary

- We create a semi-supervised framework for classifying short-text (social media text, product reviews and news comments).
- It can be used with any predictive training model, but it increases performance when training models learn from context.
- Certain size of seed (min ~ 5000 labeled examples) is still required. There has to be enough context in the examples and some overlap in context in between samples.
- We propose a way to create a seed fast using users profiles.
- For seed size of 5000, SLP with fastText performs the best in all cases.
- For seed size of 50000, SLP with VDCNN performs the best in all cases.