Carnegie Mellon University

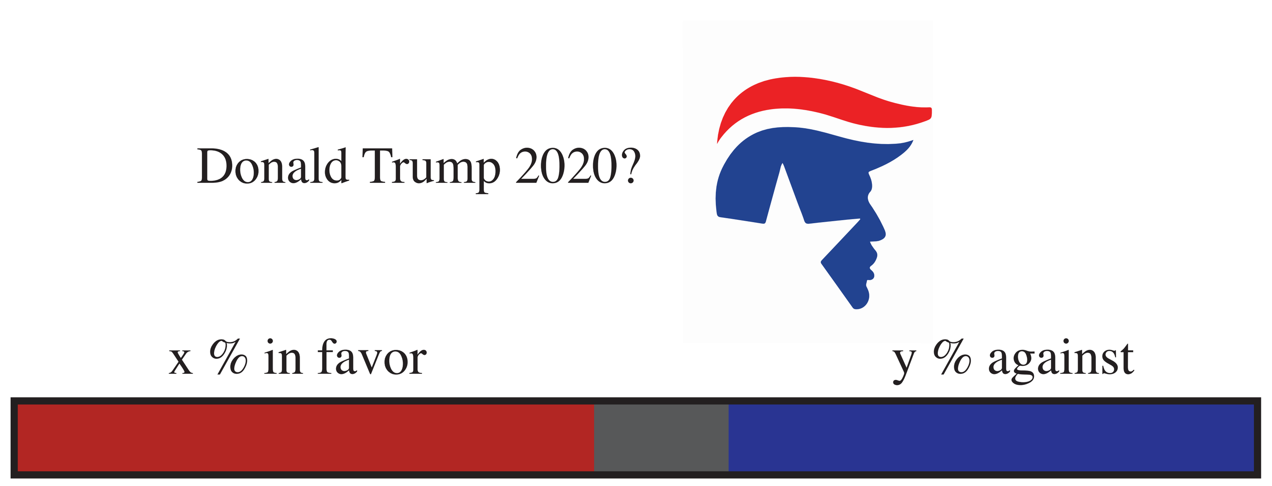
Semi-Supervised Stance Detection In Tweets

Aditya Agarwal, Sarthak Ahuja, Tanmay Agarwal
(adityaa, sarthaka, tanmaya)

ML 10-701
Spring 2019

Motivation

With widespread use of social media, it has become practice for people to express opinion online on platforms such as Twitter, Instagram, etc on important issues related to law and politics.



Automatically detecting stance from text posted on social media platforms will offer an unbiased and more accurate overview of stance of a large number of users

Existing Methods

- Several approaches apply heuristic based semi-supervised methods by using unlabelled data alongside labelled data independently.
- Large sets of unlabelled data are relatively easier to obtain and are primarily used to inform the choice of representation.
- In the context of stance detection existing methods primarily use unlabelled data to extract useful word embeddings for the labelled data and follow it up with a supervised learning approach -

	Embedding Method	**Learning Classifier**
Zarella et al. [1]	Skip-Gram	RNN
Wei et al. [2]	Word2Vec	CNN
Tutek et al. [3]	Word2Vec	Ensemble (RF, GB, LR, SVM)
Liu et al. [4]	Word2Vec	Ensemble (RF, DT, SVM)
Augenstein et al. [5]	Auto-Encoder	Logistic Regression

A limitation of existing methods is that the embeddings are not interpretable

Dataset and Preprocessing

50,000 Unlabelled

SemEval 2016 Challenge*
Target: "*Donald Trump*"

707 Labelled Tweets

	AGAINST	**FAVOUR**	**NONE**
	299	260	148
dataset balancing via upsampling	299	299	299

FAVOR *Considering the fact that Bush was a president of this country, I don't see it a joke that Trump is running!*
NONE *Honestly I am gonna watch #Univision so much more now, just to support the network against #SemST*
AGAINST *@realDonaldTrump should've kept his mouth shut & not run for Pres. He is making the biggest fool out of himself. He's fired #SemST* ...

Training Set: 627 Labelled Tweets + 50,000 Unlabelled Tweets
Testing Set: 270 Labelled Tweets

Twitter data has some unique specific traits -

- 140 character limit **- use of inconsistent english** **- slangs words**

We perform the following preprocessing NLP pipeline to clean the data -

Stop Word Removal (NLTK)	Remove Special Symbols
Lower Case	Lemmatize (spaCy)
Spell Check (pyenchant)	Slang Substitution (noslang.com)

*http://alt.qcri.org/semeval2016/task6/

Methodology

Baseline -

LDA (Latent Dirichilet Allocation)

LDA[6] is a generative probabilistic model of a corpus of documents. Each document is represented as a distribution of latent topics, and each topic is represented as a distribution over words.

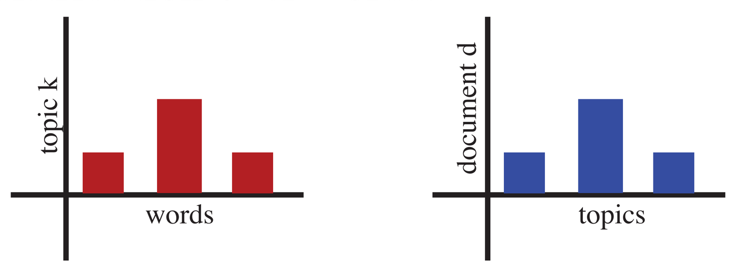


Figure 1: In LDA, the prior probability distribution of topics for a document d, P(θ_d) is modelled as a dirichilet vector α of size (#topics) and similarly the prior probability distribution of words for a given topic k, P(z_k) is modelled as a vector β of length #words. The key problem that LDA solves is of computing the posterior distribution of the hidden variables given the document -

$$P(\theta, z|d, \alpha, \beta) = \frac{P(\theta, z, d|\alpha, \beta)}{P(d|\alpha, \beta)} \quad (1)$$

We use a computationally optimized implementation of Gibbs Sampling, Mallet[8] to implement LDA. Post grid-search #topics used was 44.

Example Doc Vector - [0, ..., 0, 0.8, 0.2] (probability distribution over topics)

Pros:
Sparse representation ~ Interpretible

Para2Vec

Para2Vec[7] is a natural extension to Word2Vec, that is able to generate vector embeddings for a document of words.

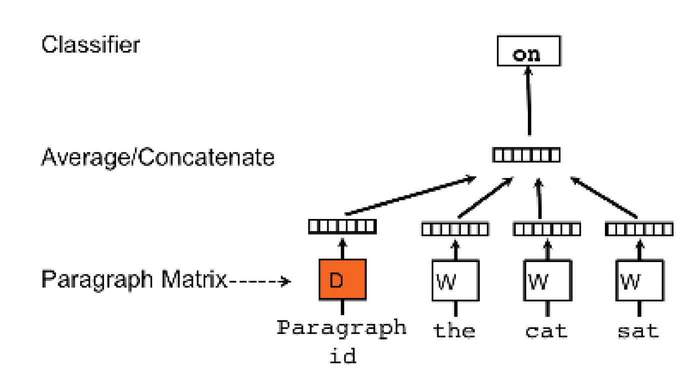


Figure 2: In Word2Vec, the context of the words in a window (the cat sat) is used to predict the next word. This forms a word matrix W of size - (#words, #hidden units) and provides us the required word embedding post training. Para2Vec adds an additional matrix D of size - (#paragraphs, #hidden units) indexed by paragraph tokens. Collectively, appending or averaging, the paragraph and word vectors, are used to predict the next word. Inherently, D acts as a topic memory and post training learns vector represention of the paragraphs. [7]

We use GenSim[9] Doc2Vec to train our model for 40 epochs on our training set. Post grid-search, #features used was 100.

Example Doc Vector - [0.73, ..., 1.1, 2.3]

Pros:
Captures sequential nature of the text.
Locally + globally coherent embeddings

Experimental Approach - LDA2Vec

For stance detection we would like to have a tweet representation that offers all the aforementioned pros. Hence, for our ML course project we choose to implement a hybird approach, LDA2Vec[10] that combines our 2 baseline approaches and offers their collective benefits. The model resembles the architecture of Para2Vec and can be summarized as follows -

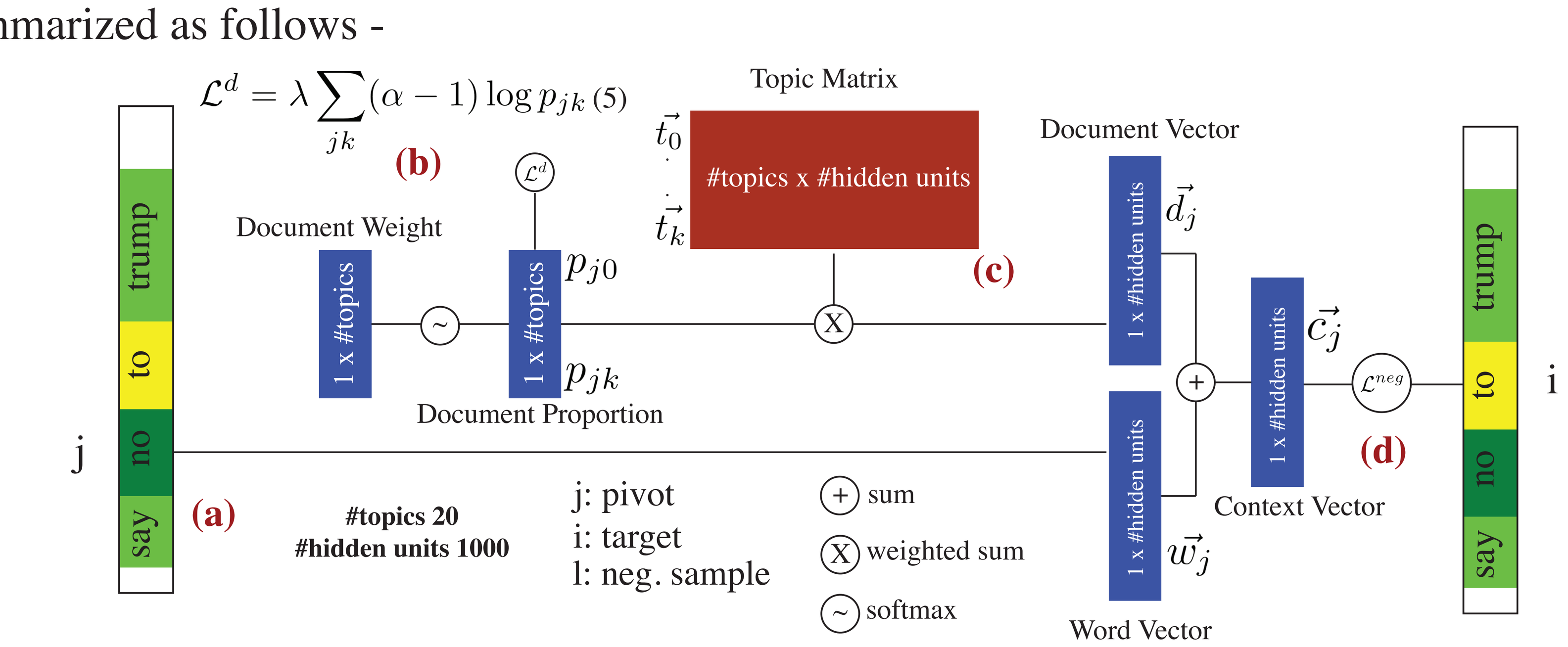


Figure 3: LDA2Vec Pipeline: **(a)** A sliding window runs across the input text and a pivot word is selected, indexed by j in this case, and passed to a linear layer of hidden units. **(b)** A randomly initialized document weight vector is initialized and converted to a probability distribution by passing it through a softmax function. Inspired by LDA, the vector is sparsified by using a dirichilet loss function (5), we set lambda to 100, and alpha to 1/20 **(c)** A topic matrix is initialized with Vanilla LDA and a document vector is created using a weighted sum of the topics. **(d)** The final loss function is a negative sampling loss function as described below where n is the number (15) of random negative samples used.

Document And Context Vectors

$$\vec{d_j} = p_{j0}.\vec{t_0} + ... + p_{jk}.\vec{t_k}, 0 \leq p_{jk} \leq 1 \quad (2)$$

$$\vec{c_j} = \vec{w_j} + \vec{d_j} \quad (3)$$

Loss Function Definition

$$\mathcal{L}^{neg}_{ij} = \log \sigma(\vec{c_j}.\vec{w_i}) + \Sigma_{l=0}^{n} \log \sigma(-\vec{c_j}.\vec{w_l}) \quad (4)$$

$$\mathcal{L} = \mathcal{L}^d + \Sigma_{ij} \mathcal{L}^{neg}_{ij} \quad (6)$$

Results

Pipeline -

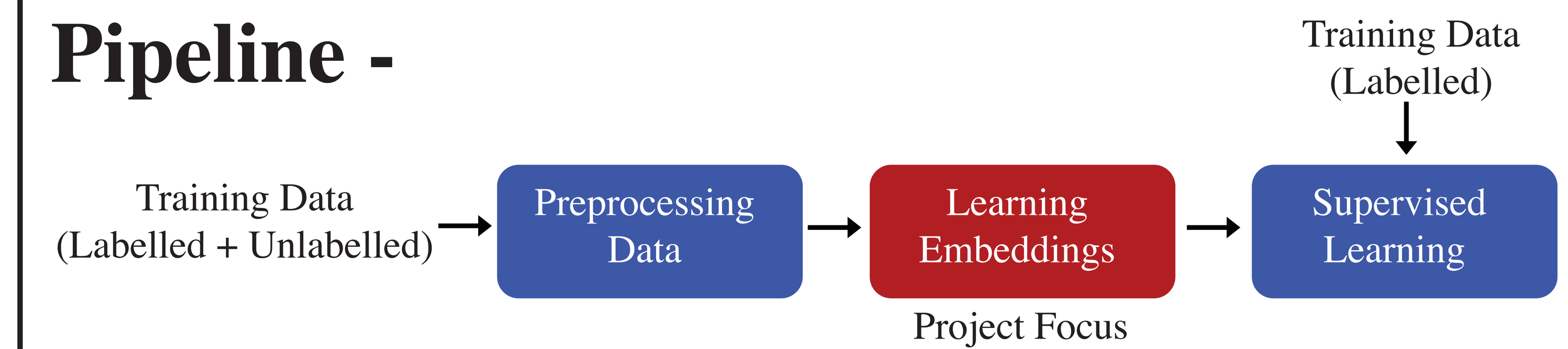
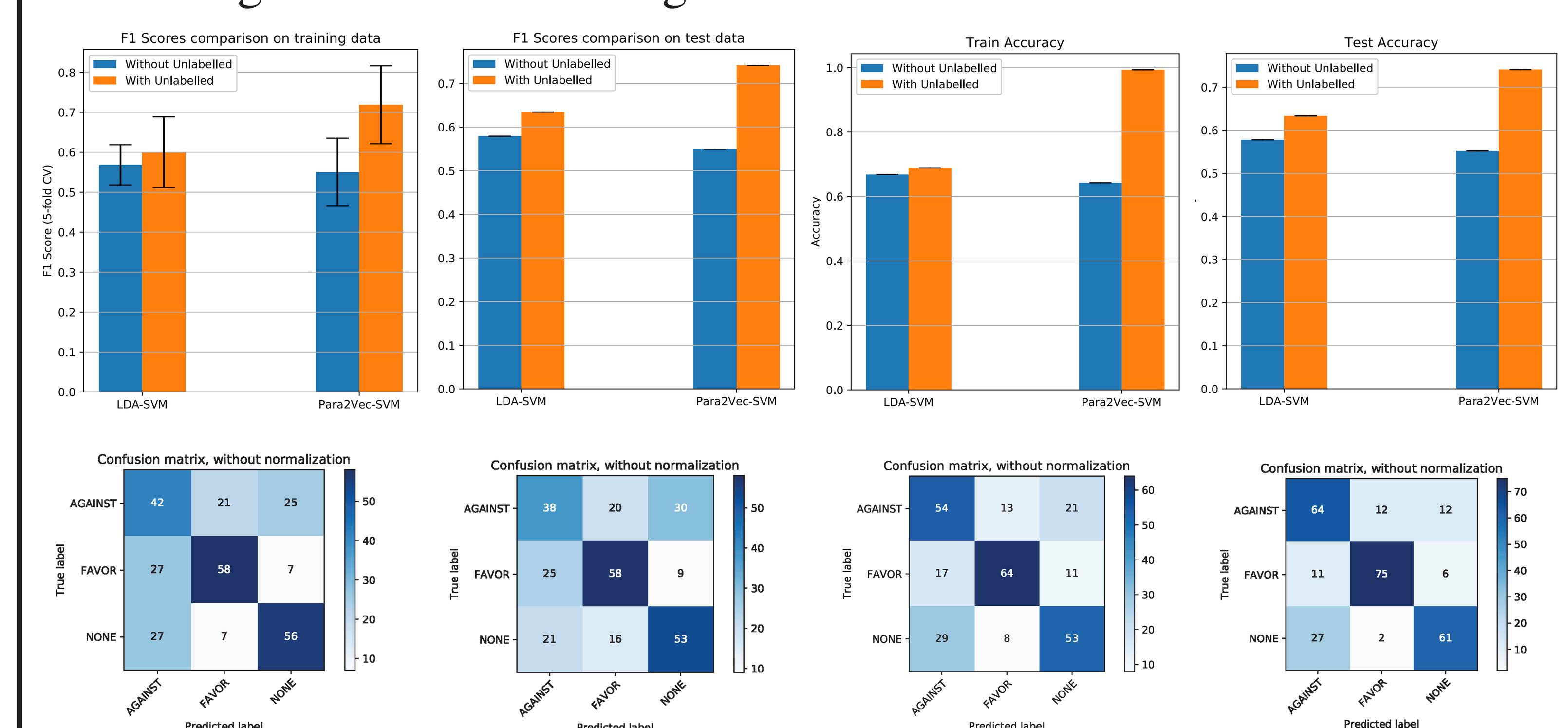


Figure 4: We use the standard pipeline and use our generated embeddings (informed by unlabelled data) to train a supervised classifier. For this project we use a C-SVM as our standard classifier and keep it constant across our experiments. We perform grid-search over values of C and the kernel to be used to get the best parameters in each case - LDA w.o. Unlabelled (C = 1, linear kernel), LDA w. Unlabelled (C = 1000, linear kernel), Para2Vec w.o. Unlabelled (C = 1000, linear kernel), Para2Vec (C = 1000, RBF Kernel)

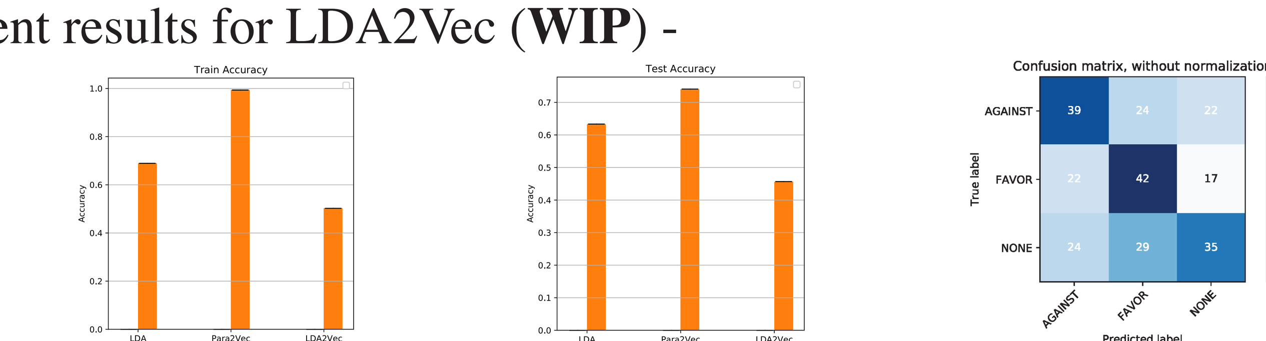
Evaluating the benefits of using unlabelled data -



LDA without Unlabelled | LDA with Unlabelled | Para2Vec without Unlabelled | Para2Vec with Unlabelled

Figure 5: .(above) F1-score and accuracy plots that justify for both our baselines, that adding the unlabelled data to the embedding process improves the overall accuracy of the classifier on both the training and the testing set. (below) The corresponding confusion matrices elucidate more information about the per-class accuracy.

Current results for LDA2Vec (**WIP**) -



Comparing the topics generated by LDA and LDA2Vec -

Topic 1	**Topic2**	**Topic 3**
illegal	people	mexican
immigrant	american	call
immigration	white	immigrant
kill	black	rapist
woman	man	criminal
rap	learn	comment
crime	understand	sell
family	hat	drug
murder	care	butt_plug
rape	realize	drug_dealer

Topic 1	**Topic 2**	**Topic 3**
come	missusa	great
border	univision	poll
escape	pull	candidate
build	dump	right
murder	usa	party
mexico	comment	presidential
kill	miss	republican
illegal	drop	lead
drug	pageant	need
immigrant	macys	real

Figure 6: Analyzing the topics generated by LDA (Left) and LDA2Vec (Right) (We fetch the top-10 words for each topic embeddings)

Conclusion and Observations -

1. We can conclude that adding unlabelled data vastly improves the performance of classifiers by ~6% for LDA and ~20% for Para2Vec. Overall Para2Vec seems to perform better than the Vanilla LDA.
2. While we are able to obtain a similar quality of topics with LDA2Vec as compared to LDA, the generated embeddings do not reflect the expected classification quality compared to Para2Vec.

References

[1] Guido Zarrella and Amy Marsh. MITRE at SemEval-2016 Task 6: Transfer Learning for StanceDetection. Technical report.
[2] Wan Wei, Xiao Zhang, Xuqin Liu, Wei Chen, and Tengjiao Wang. A Specific ConvolutionalNeural Network System for Effective Stance Detection. Technical Report 6
[3] Martin Tutek, Ivan Sekuli, Paula Gombar, Ivan Paljak, Filip Boltuži, Mladen Karan, DomagojAlagi, and Jaˇn Snajder. TakeLab at SemEval-2016 Task 6: Stance Classification in TweetsUsing a Genetic Algorithm Based Ensemble. Technical report.
[4] Can Liu, Wen Li, Bradford Demarest, Yue Chen, Sara Couture, Daniel Dakota, Nikita Haduong,Noah Kaufman, Andrew Lamont, Manan Pancholi, Kenneth Steimel, and Sandra K Ubler.IUCL at SemEval-2016 Task 6: An Ensemble Model for Stance Detection in Twitter. Technicalreport.
[5] Isabelle Augenstein, Andreas Vlachos, and Kalina Bontcheva. USFD at SemEval-2016 Task 6:Any-Target Stance Detection on Twitter with Autoencoders. Technical report.
[6] Blei DM, Ng AY, Jordan MI. Latent dirichlet allocation. Journal of machine Learning research. 2003;3(Jan):993-1022.
[7] Le Q, Mikolov T. Distributed representations of sentences and documents. InInternational conference on machine learning 2014 Jan 27 (pp. 1188-1196).
[8] Andrew Kachites McCallum. Mallet: A machine learning for language toolkit. 2002.
[9] Radim Rehurek and Petr Sojka. Software framework for topic modelling with large corpora. InIn Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks. Citeseer,2010.
[10] Christopher E Moody. Mixing dirichlet topic models and word embeddings to make lda2vec.arXiv preprint arXiv:1605.02019, 2016.