Novel approaches to sentiment analysis for stock prediction

Chris Wang, Yilun Xu, Qingyang Wang
{chrwang, ylxu, iriswang} @ stanford.edu

Introduction

- **Efficient-market hypothesis**: market reflects ALL available info
  - We **don't agree**--there may be different interpretations of info
- Fundamental analysis, technical analysis, and machine learning
  - Obviously, we will use machine learning--technical indicators can be included as features, add in company info through **news**!
    - ✓ News is reflective of company fundamentals, public mood
- Existing models are limited in architecture and features used -> try a wide scope of features and methods

Data Set

Trading and news data of 20 NASDAQ companies from 2013 to 2017, with ~24K obs. (~16% as test) and ~70 features (one hot encoded):

Trading	Daily trading volume and price from Yahoo API	**Macro**	GDP, CPI and Libor from Fed database
News	Ticker-specific news scraped from **Google** and **NY Times**	**Technical**	Self-constructed CCI, RSI and EVM

Richer and more meaningful news sources for real-life applications

Label design:
- Y = 1 if adj. close price >= last adj. close
- Y = 0 if adj. close price < last adj. close

Data visualization shows the complicated relationship between stock movement and the selected features, indicating that common classification approaches may not work

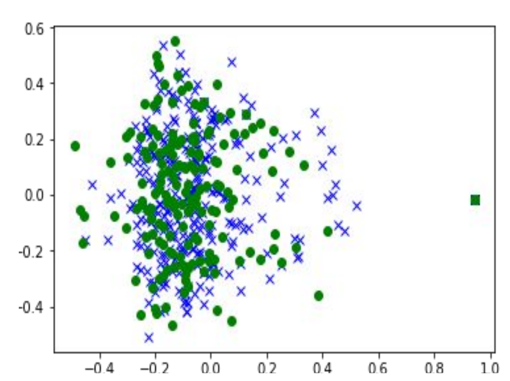
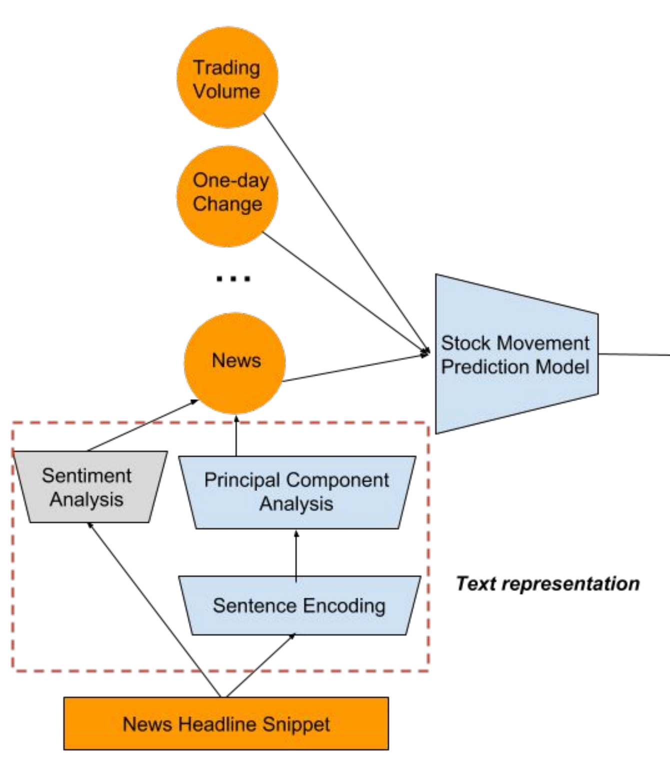


Fig1: Label vs. two principal components of news

Model Overview

Goal: supervised learning problem to predict the next-day stock movement, Y, based on numerical features and the news text



- the news text is used to get sentiment feature or represented as fixed-length (512) encoded vector as in **text representation**
  - if length 512 vector, **Principal Component Analysis** is applied to the vector (reduces dimension to 20)
- all features are then used in one of the **stock movement prediction** models

Text Representation

We incorporate the news text as a feature in our prediction model in two ways.

Get the sentiment first using a separate technique and using it as feature
- LSTM: word2vec and LSTM (word sequential) to predict direction of sentiment
- R Package: presence of financial library keywords, dictionary based method

Convert sentences to fixed-length integer vectors using encoding methods, and use each dimension of the vector as an input feature.

The goal of most sentence embedding methods is to capture similarity between vectors using orderings of characters/words/sentences (see table)

- Models are pretrained on a large corpus of sentences, "transfer learning"
- Approaches using words, then avg.:
  - word2Vec: bag-of-Words, skip-Grams
  - ELMo: internal states of word bidirectional LSTM
  - FastText: based on character seqs, not words
- Entire Sentence Encoders:
  - Skip-Thoughts: encoder-decoder with sentences
  - Google USE: deep average network encoder, supports a variety of data types

Encoding of the entire sentence, large pretrained corpus, and dexterity with data types

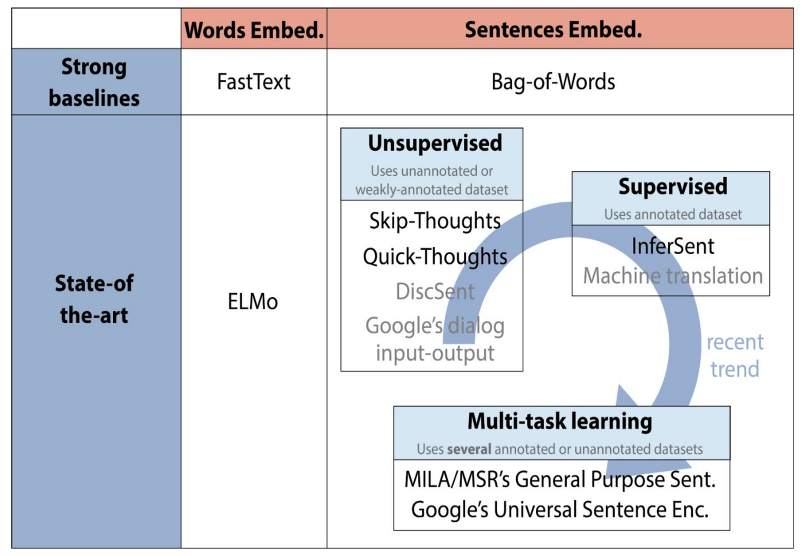


Table 1: approaches of embedding models

Stock Movement Prediction

Logistic regression *(LR, baseline models)* with or without sentiment features

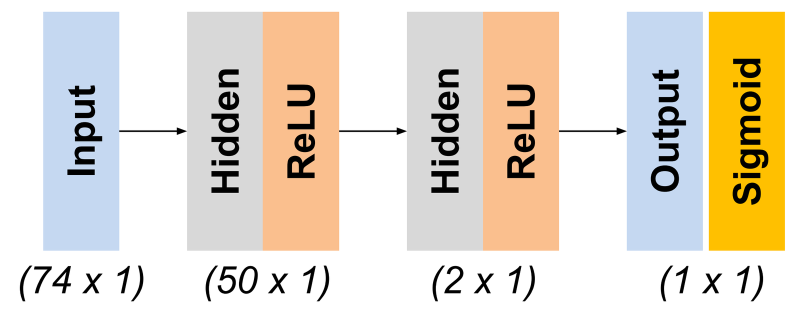
Random Forest with cross-entropy loss. Tune max depth and max features to control overfitting / underfitting

SVM:
- As shown in previous research, SVM tends to be effective in stock prediction
- RBF kernel captures the high-dimension nature of stock movement
- Tune cost parameter to control overfitting / underfitting

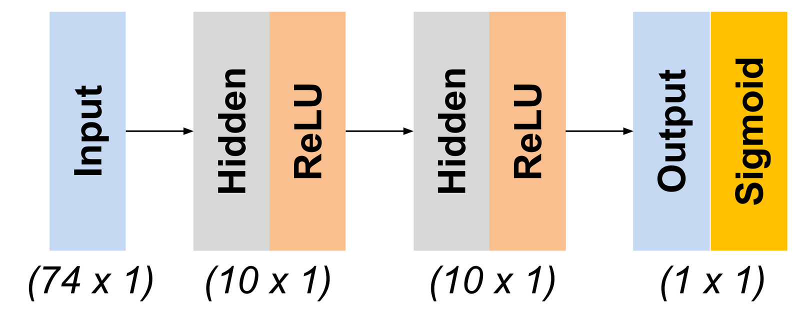
Neural Network-based Models

(a) Neural Network is constructed and tuned on two datasets (constructed using Google and NY Times news) separately

Dataset constructed with ***Google news***
Best model structure



Dataset constructed with ***NY Times***
Best model structure



(b) CNN is introduced to explore relationships between sentiment-related features (output of encoder/PCA). Two 1D-conv layers, each followed by a pooling layer, are included before the final fully connected layer

(c) RNN with one LSTM layer is performed on subsets constructed with data from each ticker to capture the time series nature of stock movement

Results and Analysis

Model	Performance on Google news		Performance on NY Times	
	Training accuracy	Test accuracy	Training accuracy	Test accuracy
LR w/o sentiment	*0.5280*	*0.5169*	*0.5280*	*0.5169*
LR w/ sentiment	*0.5337*	*0.5046*	*0.5308*	*0.5185*
Random Forest	*0.7770*	*0.4870*	*0.7601*	*0.5006*
SVM	***0.5650***	***0.5430***	***0.6005***	***0.5414***
NN	*0.6172*	*0.5273*	*0.5881*	*0.5259*
CNN	*0.5816*	*0.5100*	*0.5464*	*0.5204*
RNN	*0.4931*	*0.4809*	*0.4832*	*0.4695*

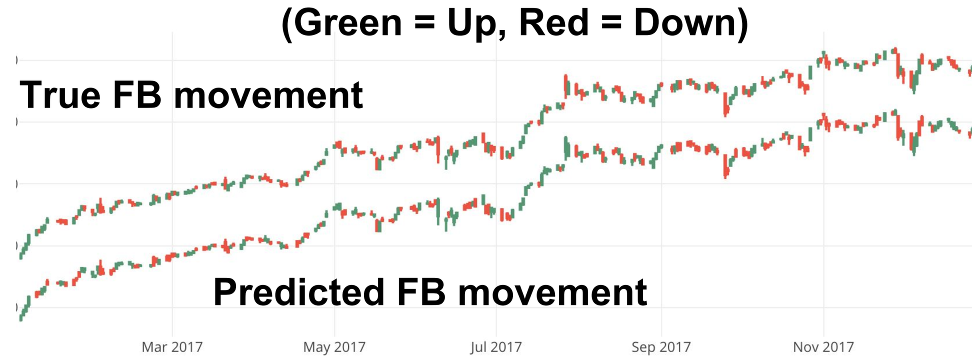
- **SVM** has the best performance on both data sets, with relatively balanced prediction on (+) and (-) labels. RBF kernel (infinite dimension) is efficient in stock movement separation
- **CNN/ NN** report decent results .To avoid overfitting, we limit the complexity of hidden layers => unable to capture all relationships

Bias	LR / Random Forest cannot capture the complex nature of stock movement	**Variance**	Random Forest is severely overfitting due to the model structure	**Stability**	RNN has unstable performance on each subset, due to limited data size

- Tickers with **more news** tend to achieve **higher accuracy**

(Green = Up, Red = Down)

Examining the predictions closely, we found the best model is more able to detect major up / downs than smaller changes



Future Work

- **Customize the loss function:** Most of our models are not customized to achieve a balanced performance on both (+) and (-) classes. We think customizing the loss function (e.g., using binary cross-entropy) may help us to achieve balanced performance
- **Enhance the data quality:** We built the data set using Google and NY Times news we scraped from the internet. Irrelevant news may be included. We believe manually cleaning the data or including models to check the validity of news may improve the performance

References and Sources

- Word2Vec: https://github.com/mmihaltz/word2vec-GoogleNews-vectors
- LSTM: https://github.com/clairett/pytorch-sentiment-classification
- Encoding models: http://hunterheidenreich.com/blog/comparing-sentence-embeddings/
- We would like to thank Atharva Parulekar and the entire CS 229 teaching staff for all of your support and guidance throughout the course of the project.