Use Case 1: Sentiment Classification with Word2Vec

» **Task :** determine polarity of document (*news or blog or tweet*)

Classes: positive, negative and neutral

» **Assumption**: the document contains only one opinion *(not true in many cases)*

Example:

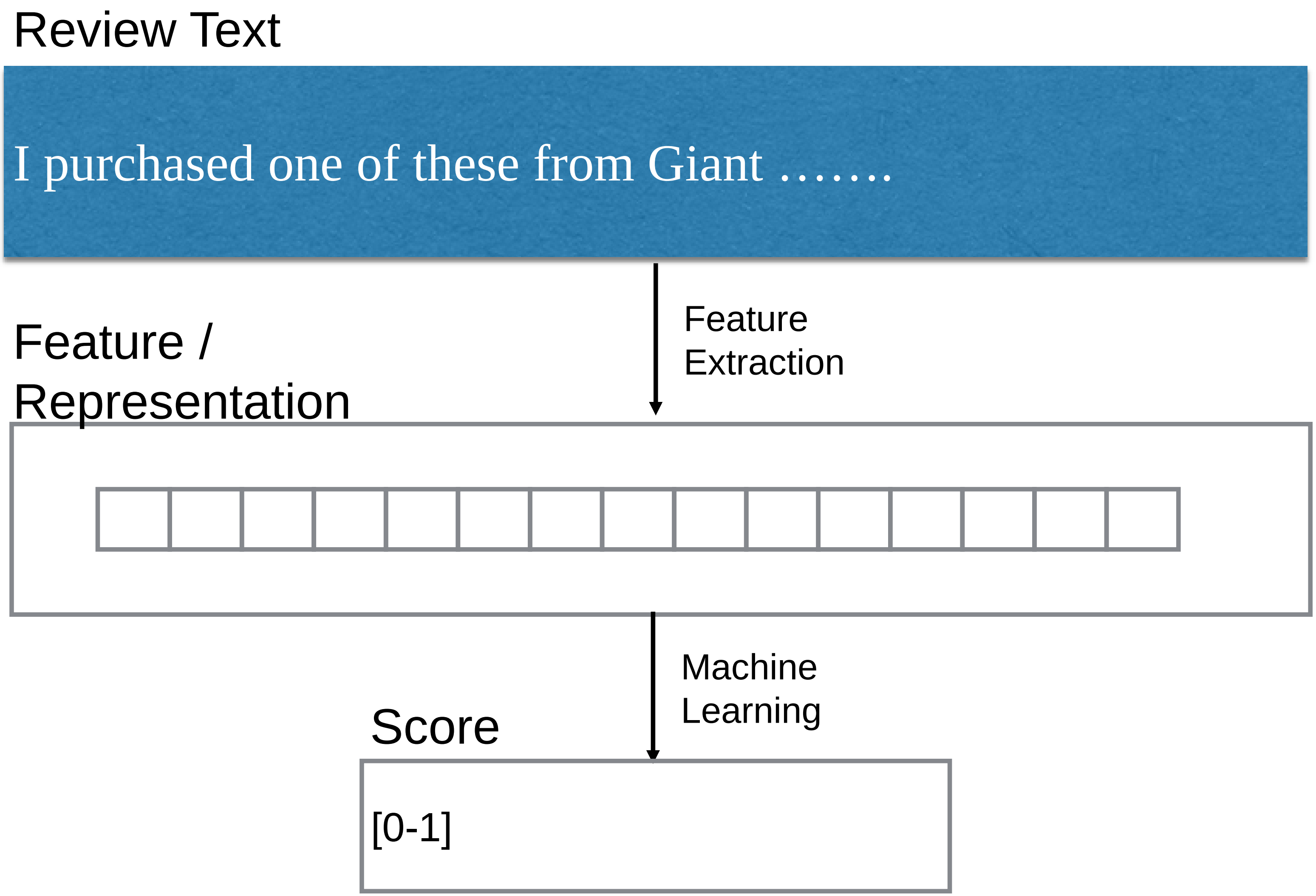
"This is a beautiful bracelet.."

Is it positive, negative or neutral?

Application Scenarios

- *Full of zany characters and richly applied satire, and some great plot twists:* is this a positive or negative review?
- Public opinion on the stock market mined from Tweets
- What do people think about a political candidate or issue?
- Can we predict election outcomes or market performance from sentiment analysis?

Overview of Approach



Review Text

I purchased one of these from Giant …….

Feature Extraction

Feature / Representation

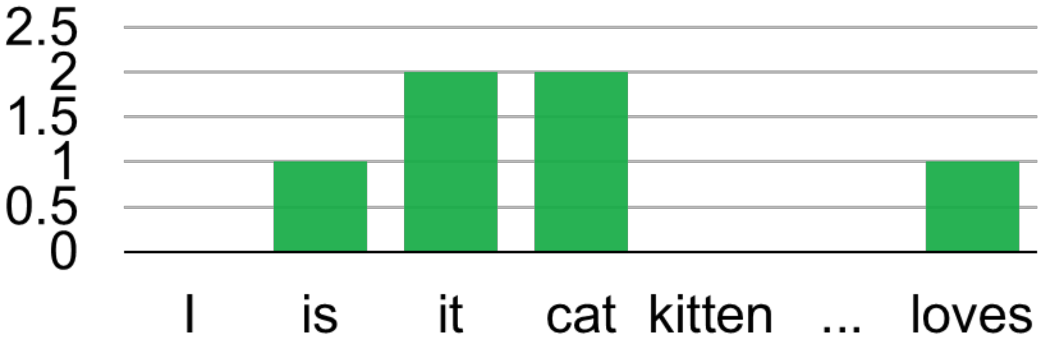
Machine Learning

Score

[0-1]

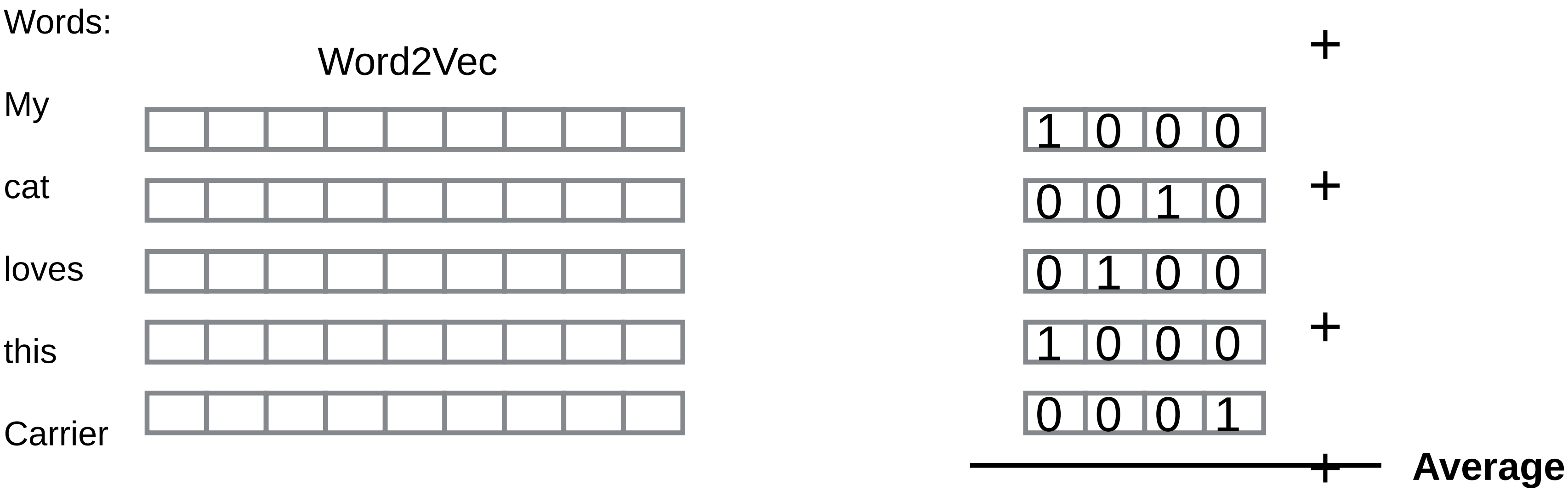
Feature Representation

- Review: *My small cat loves this carrier. It is very soft inside and it has a small window that my cat can use to look outside.*



Some are useful words and others not. How to assign weights ?

Word2Vec: skip-gram

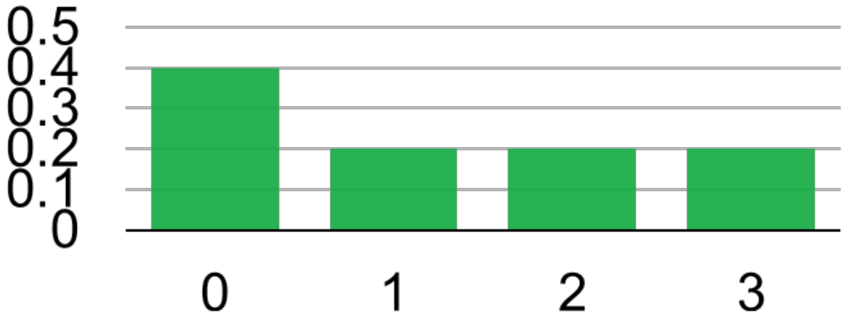


Words:

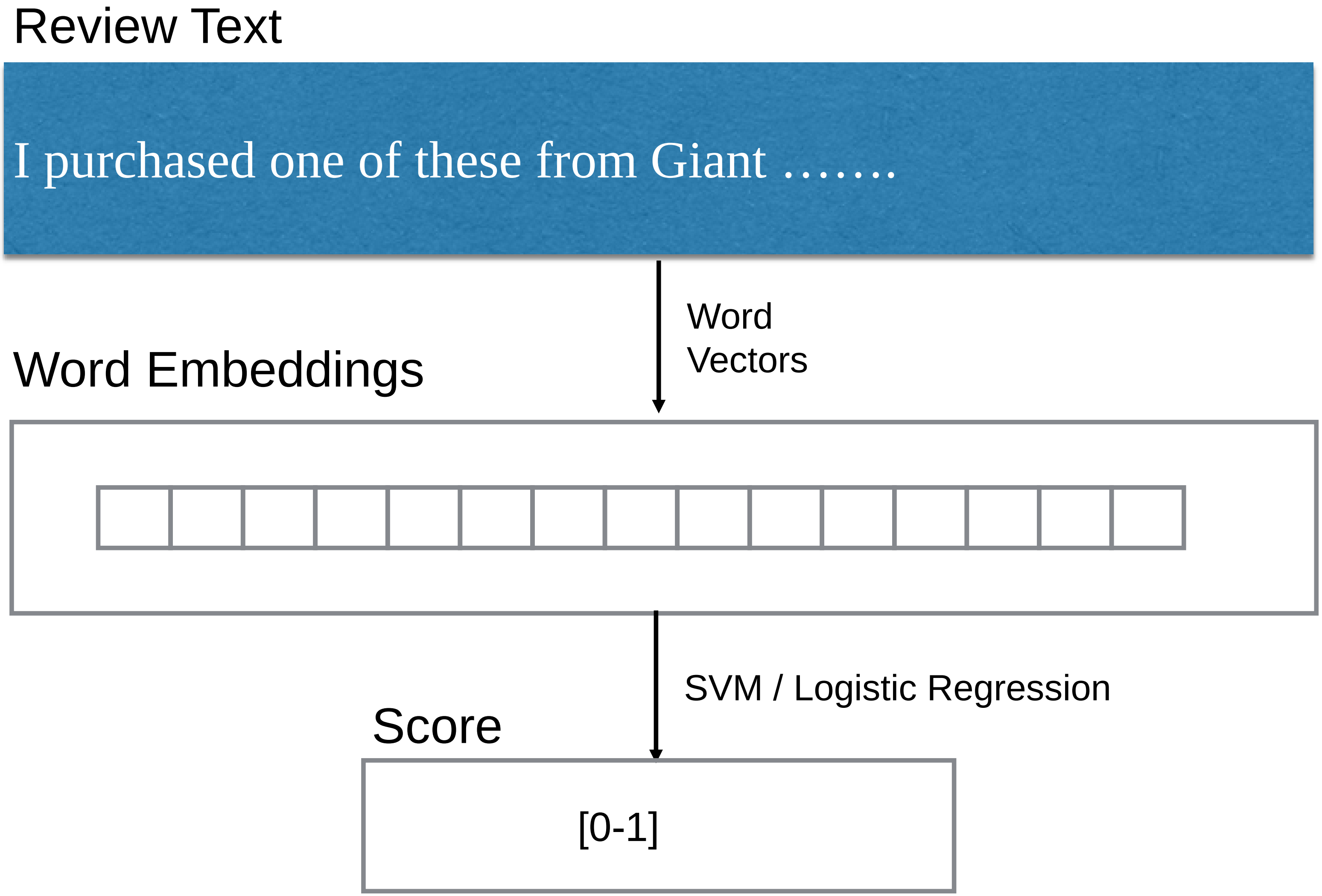
Word2Vec

Words:				
My	1	0	0	0
cat	0	0	1	0
loves	0	1	0	0
this	1	0	0	0
Carrier	0	0	0	1

Average



Approach



Review Text

I purchased one of these from Giant …….

Word Vectors

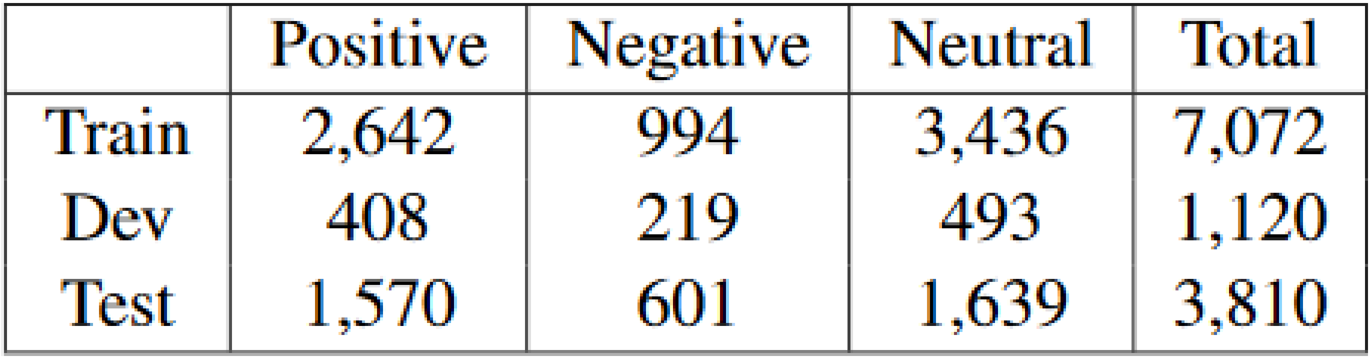
Word Embeddings

SVM / Logistic Regression

Score

[0-1]

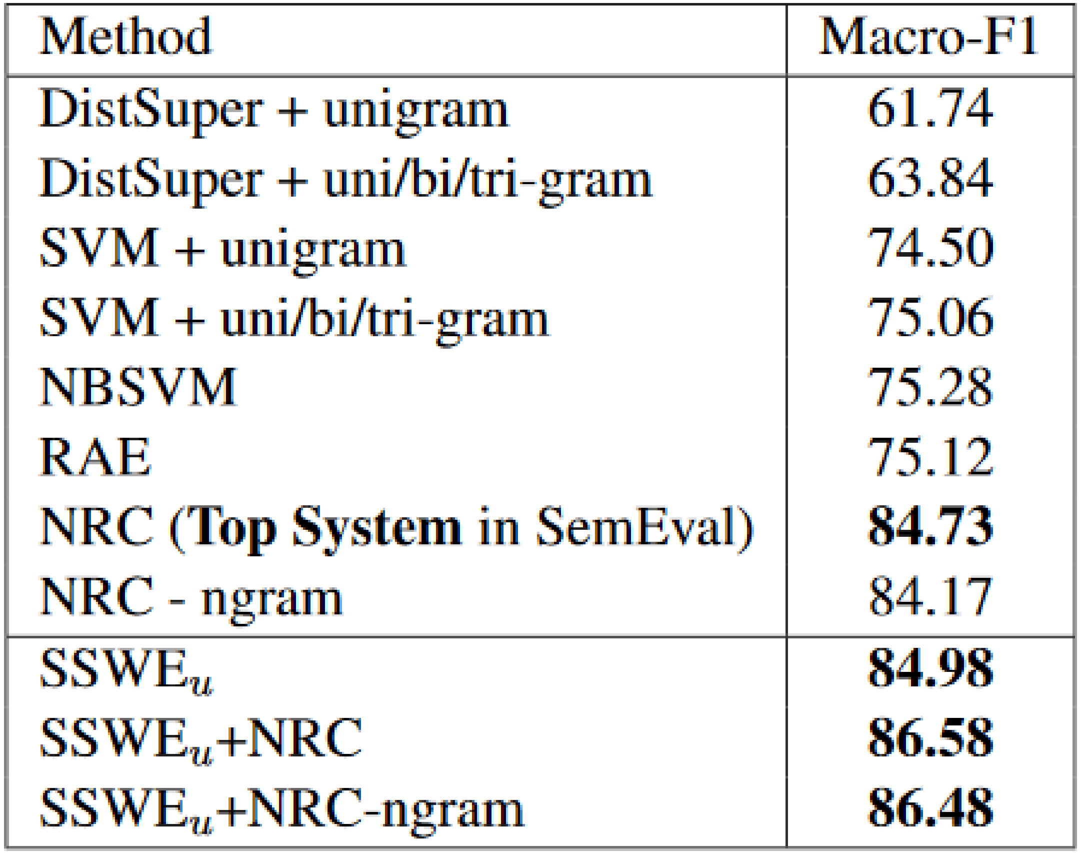
Performance Evaluation



	Positive	Negative	Neutral	Total
Train	2,642	994	3,436	7,072
Dev	408	219	493	1,120
Test	1,570	601	1,639	3,810

SemEval-13 Twitter Sentiment Classification Data

Performance Evaluation



Method	Macro-F1
DistSuper + unigram	61.74
DistSuper + uni/bi/tri-gram	63.84
SVM + unigram	74.50
SVM + uni/bi/tri-gram	75.06
NBSVM	75.28
RAE	75.12
NRC (**Top System** in SemEval)	**84.73**
NRC - ngram	84.17
SSWE_u	**84.98**
SSWE_u+NRC	**86.58**
SSWE_u+NRC-ngram	**86.48**

Advanced Feature Engineering

Lexicons + Handcoded rules

Word2Vec (Finetuned)

Word vectors can replace manual feature engineering!

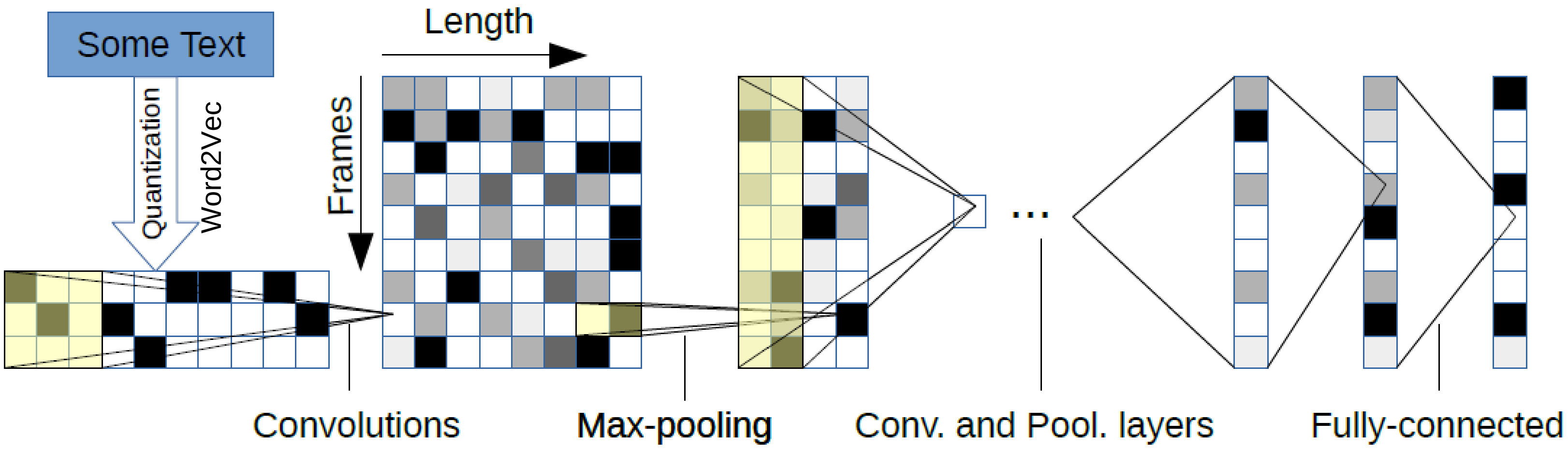
Use Case 2: Sentiment Classification with CNN

- Sometimes the classification task is tough
- **Word2Vec + SVM :** Not always achieve high performance
  - E.g. More than word context is required

CNN Approach

Text pre-processing for CNN

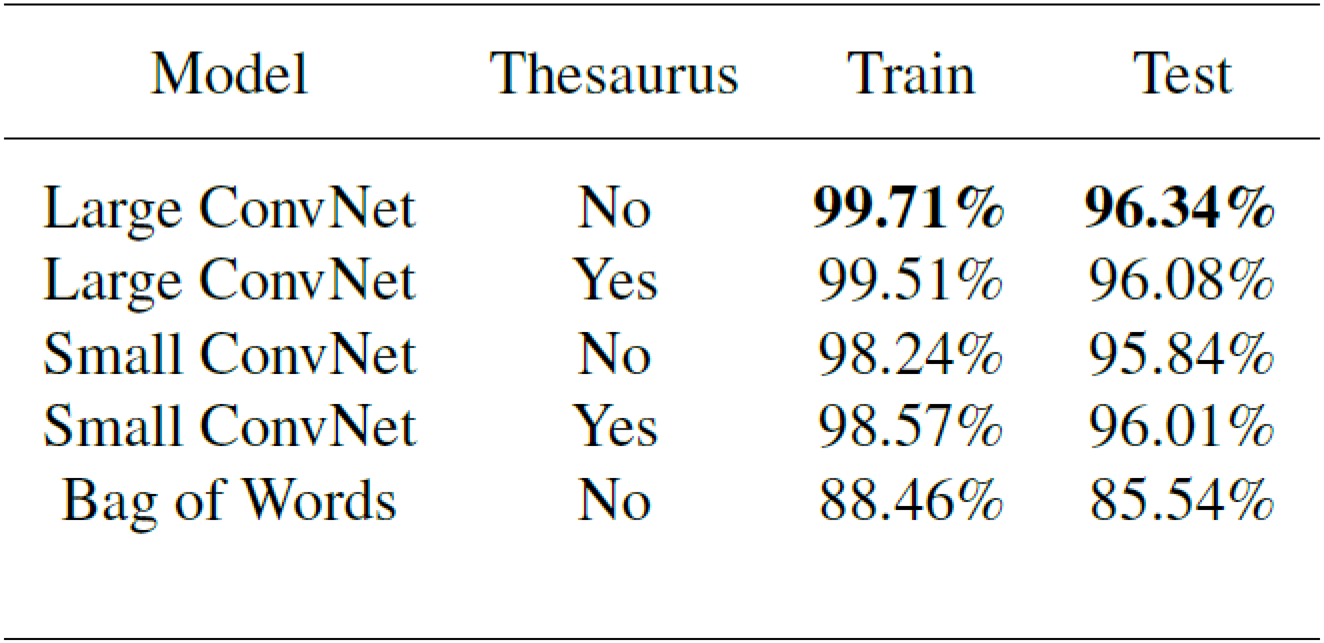
- Extract Word2Vec for each word in the text
- Keep the sequence intact
- Pass them all as a matrix to CNN



Data Augmentation

- CNN requires large number of training data
- In image classification, CNN is usually trained with translating, scaling, rotating and flipping the input images (to ensure invariance property)
- For sentiment classification, words in text are replaced with most frequent meaning found from a thesaurus (eg. WordNet)
  - » Only randomly chosen words are replaced (not all)

Classification Performance



Model	Thesaurus	Train	Test
Large ConvNet	No	**99.71%**	**96.34%**
Large ConvNet	Yes	99.51%	96.08%
Small ConvNet	No	98.24%	95.84%
Small ConvNet	Yes	98.57%	96.01%
Bag of Words	No	88.46%	85.54%