Vector Semantics

Word and Document Representation

Savas Yildirim

Table of contents

1. Distributional Theory

2. Words and Vectors

3. Pointwise Mutual Information

4. Short and dense word vectors

5. Final Remarks for document representation

Distributional Theory

Distributional Theory

- Words that occur in similar contexts tend to have similar meanings
- This link between similarity in how words are distributed and similarity in what they mean is called the ***distributional hypothesis***.
- The hypothesis was first formulated in the 1950s by linguists: mize *Joos (1950), Harris (1954), and Firth* (1957)
- They noticed that words which are synonyms (like oculist and eye-doctor) occur in the same environment

Distribution Approach

Suppose you didn't know what the Cantonese word ***ongchoi*** meant, but you do see it in the following sentences or contexts.

- Ongchoi is delicious sauteed with garlic.
- Ongchoi is superb over rice.
- Ongchoi leaves with salty sauces

And

- spinach sauteed with garlic over rice...
- chard stems and leaves are delicious...
- collard greens and other salty leafy greens

The fact that ***ongchoi*** occurs with words like rice and garlic and delicious and salty, as do words like spinach, chard, and collard greens might suggest to the reader that ongchoi is a leafy green similar to these other leafy greens.

Ong Choi

Unsupervised Learning

- The theory is based on unsupervised learning
- Finding such unsupervised ways to learn representations of the input, instead of creating representations by hand via **feature engineering**, is an important focus of recent NLP research (Bengio et al., 2013).

Similarity

- Word or sentence similarity can be measured by vector semantics: **dog/cats** are similar words.
- Phrase and sentence similarity is useful many NLP topics:
- Question answering, paraphrasing, summarisation, translation

Word Relatedness and Semantic Field

- Also called word association
- **Coffee** and **cup** is not similar but related
- **Moon** and **astronaut** are not similar
- Semantic field is a set of words which cover a particular semantic domain and bear structured relations with each other.

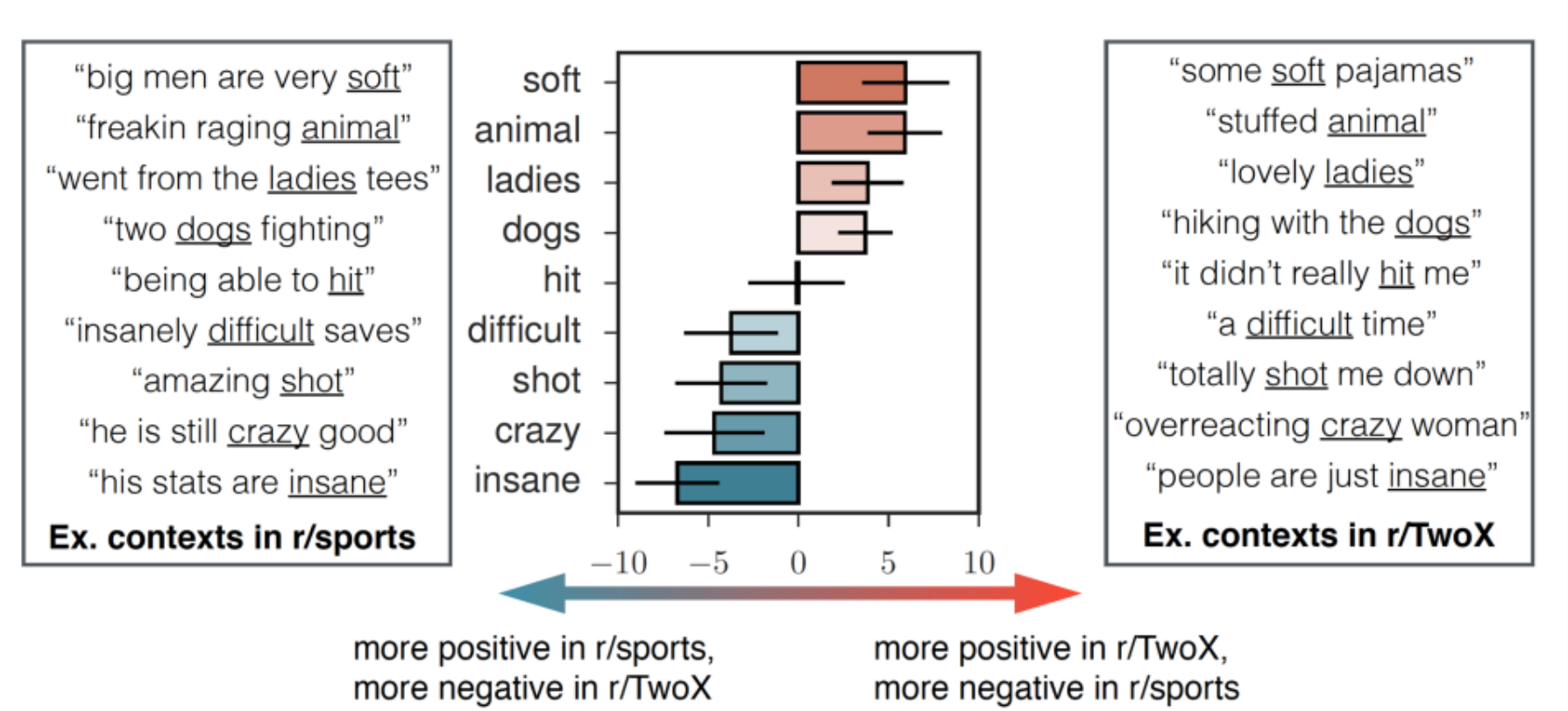
Topic Model

- Semantic fields are also related to topic models, like Latent Dirichlet Allocation, LDA (Blei et al. 2003)
- unsupervised learning on large sets of texts to induce sets of associated words from text.
- Semantic fields and topic models are a very useful tool for discovering topical structure in documents.

Connotation and Sentiment Analysis

- For example some words have positive connotations (**happy**) while others have negative connotations (**sad**).
- Some words describe positive evaluation (**great, love**) and others negative evaluation (t**errible, hate**).
- Positive or negative evaluation expressed through language is called sentiment

Frame Title

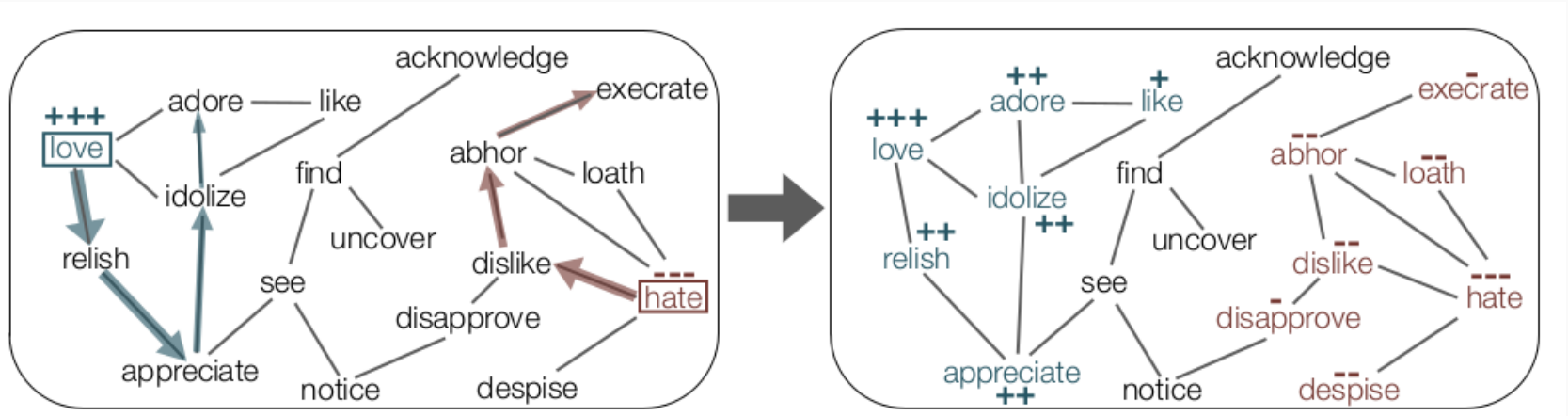


Inducing Domain-Specific Sentiment Lexicons from Unlabeled Corpora, (Hamilton et al.), 2016

Sentiment Lexicon Building

Domain	Positive seed words	Negative seed words
Standard English	good, lovely, excellent, fortunate, pleasant, delightful, perfect, loved, love, happy	bad, horrible, poor, unfortunate, unpleasant, disgusting, evil, hated, hate, unhappy
Finance	successful, excellent, profit, beneficial, improving, improved, success, gains, positive	negligent, loss, volatile, wrong, losses, damages, bad, litigation, failure, down, negative
Twitter	love, loved, loves, awesome, nice, amazing, best, fantastic, correct, happy	hate, hated, hates, terrible, nasty, awful, worst, horrible, wrong, sad

Sent Prob Algorithm



The model learns embeddings with PPMI and employed an SVD-based method to construct the word-vectors. Word vectors are then transformed a similarity connected graph.

Words and Vectors

term document matrix

- In a term-document matrix, each row represents a word in the vocabulary and each column represents a document
- It is first defined as part of the vector space model (VSM) of information retrieval (Salton, 1971).
- This representation is also called one-hot representation or Bag-of-words approach (BOW).

	As You Like It	Twelfth Night	Julius Caesar	Henry V
battle	1	0	7	13
good	114	80	62	89
fool	36	58	1	4
wit	20	15	2	3

Alternatives for Word Representation

Here are windows surrounding four sample words from the Brown corpus:

sugar, a sliced lemon, a tablespoonful of	**apricot**	jam, a pinch each of,
their enjoyment. Cautiously she sampled her first	**pineapple**	and another fruit whose taste she likened
well suited to programming on the digital	**computer**.	In finding the optimal R-stage policy from
for the purpose of gathering data and	**information**	necessary for the study authorized in the

	aardvark	...	**computer**	**data**	**pinch**	**result**	**sugar**	...
apricot	0	...	0	0	1	0	1	
pineapple	0	...	0	0	1	0	1	
digital	0	...	2	1	0	1	0	
information	0	...	1	6	0	4	0	

Example for python

```
>>>
>>>
>>>
>>>
>>> from nltk.corpus import brown
>>>
>>> text=nltk.Text(brown.words())
>>> text.concordance("apple")
Displaying 9 of 9 matches:
 have now climbed to this height . ) Apple trees grew there also . Though
s thereabouts preferred to get their apple pies at the local bakery , whi
edger , is basically this : Wright's apple pie ; ; peel , core , and slic
. Even less regard for mom and mom's apple pie goes with : Af In other wo
 itch was treated by applying strong apple cider in which pulverized bloo
leaf , made a pile as big as a small apple . The odor here was more power
 obtained from an upper bough of the apple tree . The primary quality of
 , when I was too small to reach the apple tree bough , to twist my knee
se is on me , seeing you fresh as an apple and me old and gray '' . `` I'
>>>
```

First Order vs Second Order Vectors

- word-word first order vector
  - ***wrote*** is a first-order associate of **book** or ***poem***
- word-word second order vector
  - if they have similar neighbors
    wrote is a second-order associate of words like ***said*** or ***remarked***.
    Event though they rarely occur in same context, they can be captured associated.
  - it is simply the sum of the word vectors of words in the window (Schütze, 1998)

Cosine Similarity

$$\text{cosine}(\vec{v},\vec{w}) = \frac{\vec{v}\cdot\vec{w}}{|\vec{v}||\vec{w}|} = \frac{\sum_{i=1}^{N} v_i w_i}{\sqrt{\sum_{i=1}^{N} v_i^2}\sqrt{\sum_{i=1}^{N} w_i^2}}$$

Likelihood ratio,Jaccard, PMI,Chi-Square, Dice, Euclidean, Manhattan are other metrics

TF-IDF Weigthing

TF: The frequency of word in a document

$$\mathrm{tf}_{t,d} = \begin{cases} 1 + \log_{10} \mathrm{count}(t,d) & \text{if } \mathrm{count}(t,d) > 0 \\ 0 & \text{otherwise} \end{cases}$$

The document frequency *DF* of a term t is simply the number of documents it occurs in

The Collection Frequency *CF* is total number of occurrence of a term t in entire corpus.

	Collection Frequency	Document Frequency
Romeo	113	1
action	113	31

$idf_t = log_{10}(\frac{N}{df_t})$
$tfidf_t = tf * idf$

Some DF and IDF value from shakepeare plays

Word	df	idf
Romeo	1	1.57
salad	2	1.27
Falstaff	4	0.967
forest	12	0.489
battle	21	0.074
fool	36	0.012
good	37	0
sweet	37	0

Sir John Falstaff is a fictional character who is mentioned in four plays by William Shakespeare and appears on stage in three of them (wikipedia)

TF-IDF transformation of plays

	As You Like It	Twelfth Night	Julius Caesar	Henry V
battle	0.074	0	0.22	0.28
good	0	0	0	0
fool	0.019	0.021	0.0036	0.0083
wit	0.049	0.044	0.018	0.022

Pointwise Mutual Information

Optional: Pointwise Mutual Information (PMI)

Alternative weighting scheme to tf-idf is PMI

It weighs the association between two words by asking how much more the two words co-occur than we would have a priori expected them to appear by chance.

$$\text{PMI}(w,c) = \log_2 \frac{P(w,c)}{P(w)P(c)}$$

where w is word in rows, c is context in colums

PPMI

PMI values range from negative to positive infinity. But negative PMI values (which imply things are co-occurring less often than we would expect by chance) tend to be unreliable unless our corpora are enormous. (Church and Hanks 1989, Dagan et al. 1993, Niwa and Nitta 1994)

$$\text{PPMI}(w,c) = \max(\log_2 \frac{P(w,c)}{P(w)P(c)}, 0)$$

Problem for PMI

PMI has the problem of being biased toward **infrequent events;** very rare words tend to have very high PMI values. One way to reduce this bias is to slightly change the computation for P(c), using a different function Pα(c) that raises contexts to the power of α

$$\text{PPMI}_\alpha(w,c) = \max(\log_2 \frac{P(w,c)}{P(w)P_\alpha(c)}, 0)$$

$$P_\alpha(c) = \frac{count(c)^\alpha}{\sum_c count(c)^\alpha}$$

Levy et al. (2015) found that a setting of α = 0.75 improved performance of embeddings on a wide range of tasks

Another Solution for rare words

Another possible solution is Laplace smoothing: Before computing PMI, **a small constant k** (values of 0.1-3 are common) is added to each of the counts, shrinking (discounting) all the non-zero values

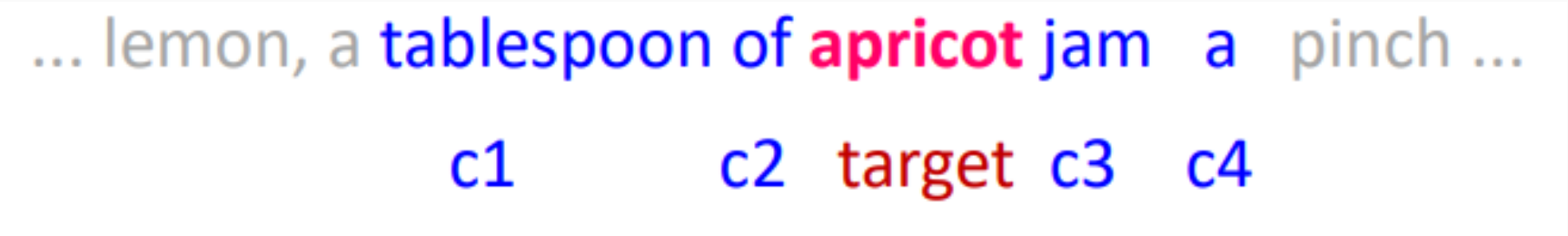
Short and dense word vectors

Word2vec and Glove

Very dense, short vectors

- The skip-gram algorithm is one of two algorithms in a software package called **word2vec** (Mikolov et al. 2013)
- The word2vec methods are found fast and efficient *https://code.google.com/archive/p/word2vec/*
- The equally popular method is **GloVe** (Pennington et al., 2014) *http://nlp.stanford.edu/projects/glove/*
- **fastText** and other
  Fasttext http://www.fasttext.cc/

Skip Gram Model



- The goal is to train a classifier such that, given a tuple (t, c) of a target word t paired with a candidate context word c (for example ***(apricot, jam)***, or perhaps **(apricot, aardvark)** it will return the probability that c is a real context word (true for jam, false for aardvark):

$P(+|t, c)$

$P(-|t, c) = 1 - P(+|t, c)$

Frame Title

$$P(+|t, c_{1:k}) = \prod_{i=1}^{k} \frac{1}{1 + e^{-t \cdot c_i}}$$

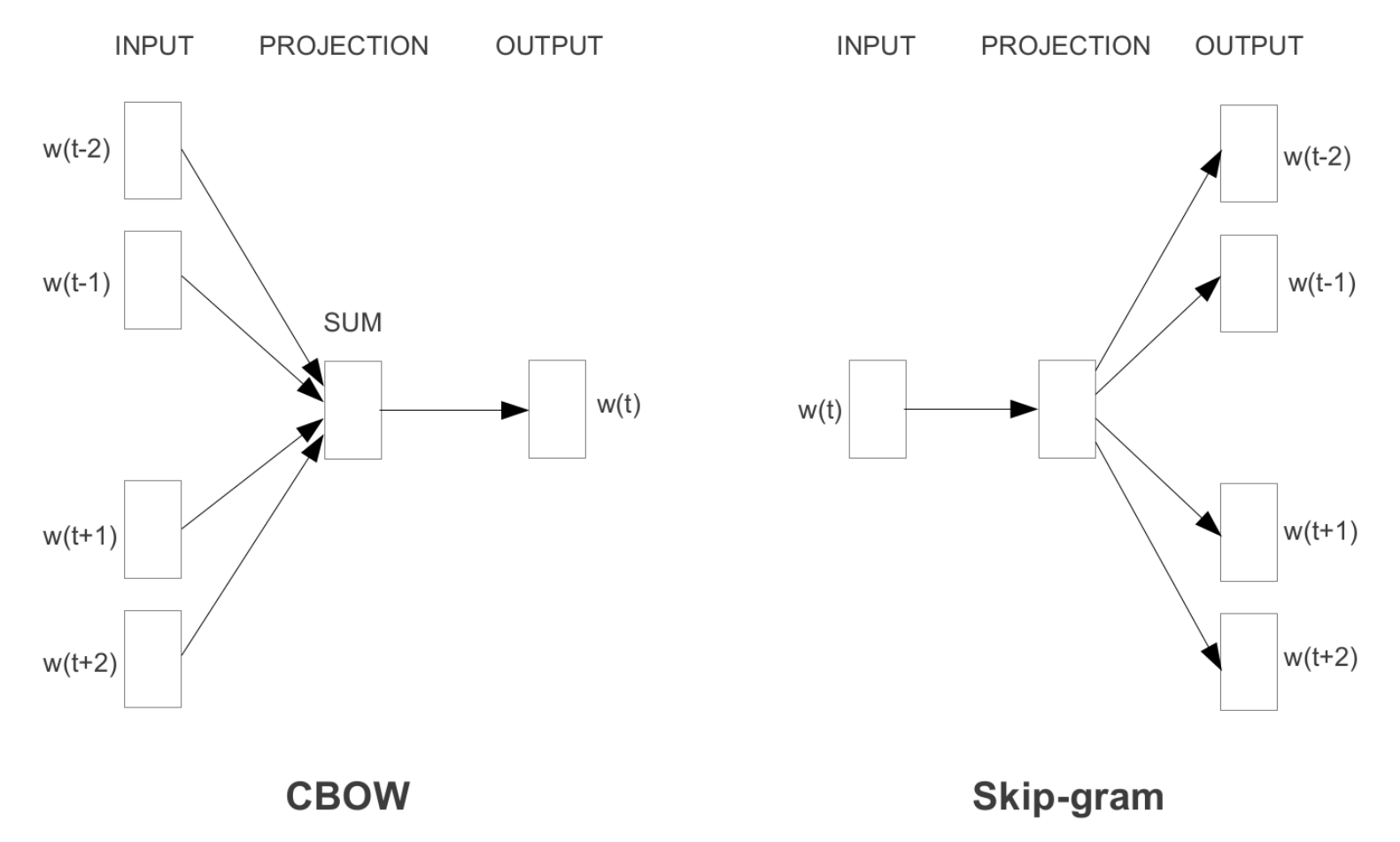
$$\log P(+|t, c_{1:k}) = \sum_{i=1}^{k} \log \frac{1}{1 + e^{-t \cdot c_i}}$$

Negative Sampling

positive examples +	
t	c
apricot	tablespoon
apricot	of
apricot	preserves
apricot	or

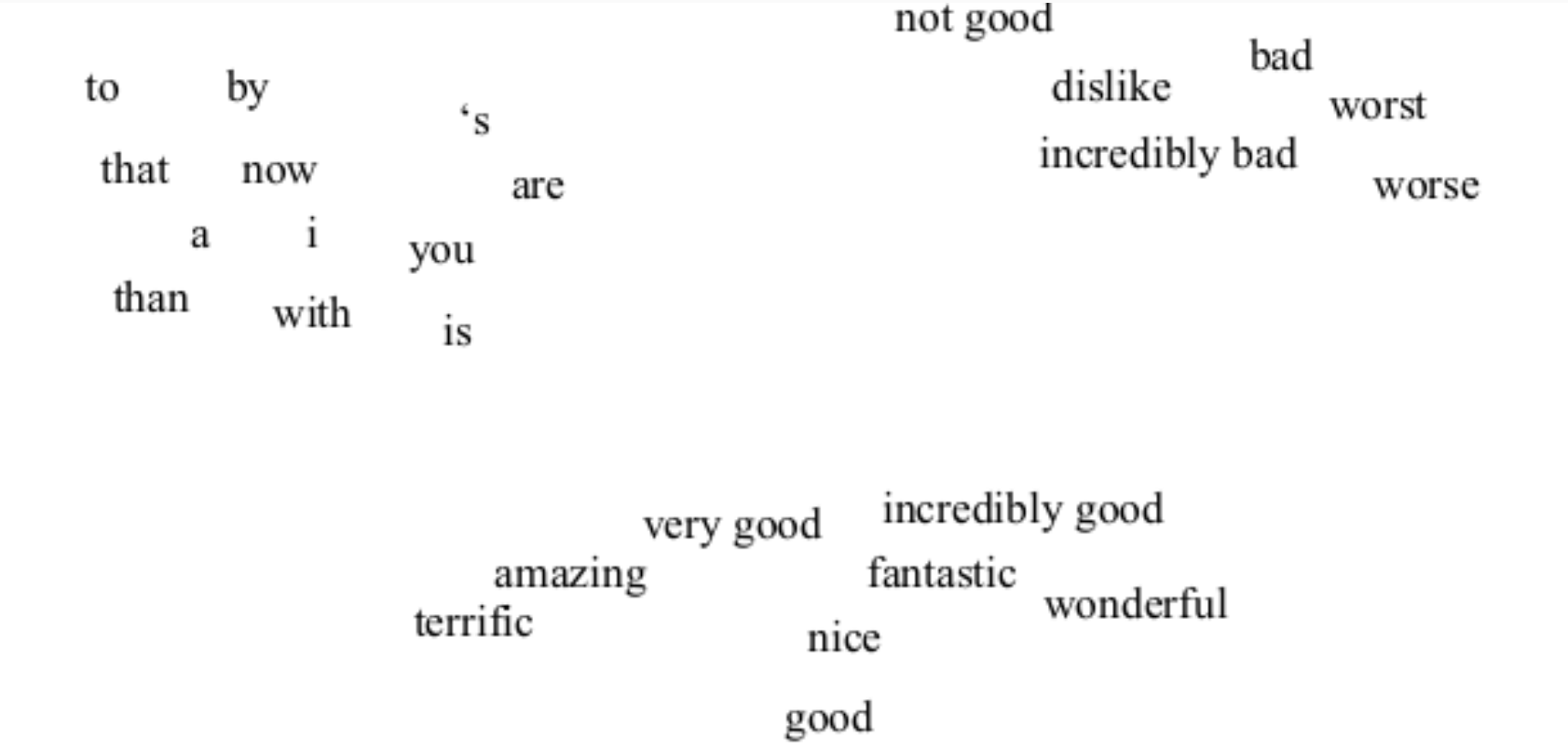
negative examples -			
t	c	t	c
apricot	aardvark	apricot	twelve
apricot	puddle	apricot	hello
apricot	where	apricot	dear
apricot	coaxial	apricot	forever

Cbow and Skip Gram

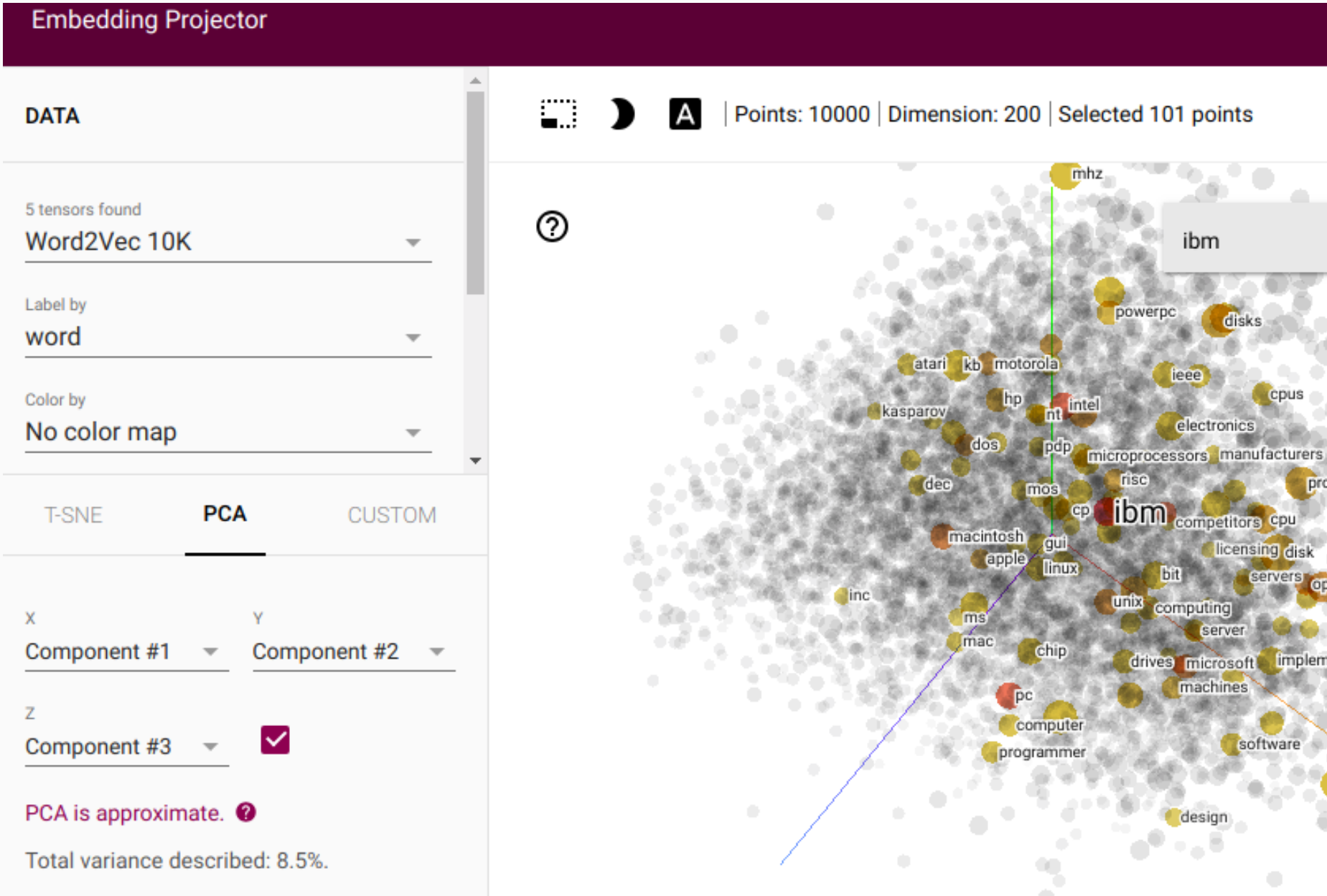


Word Mapping

t-SNE mapping of the words



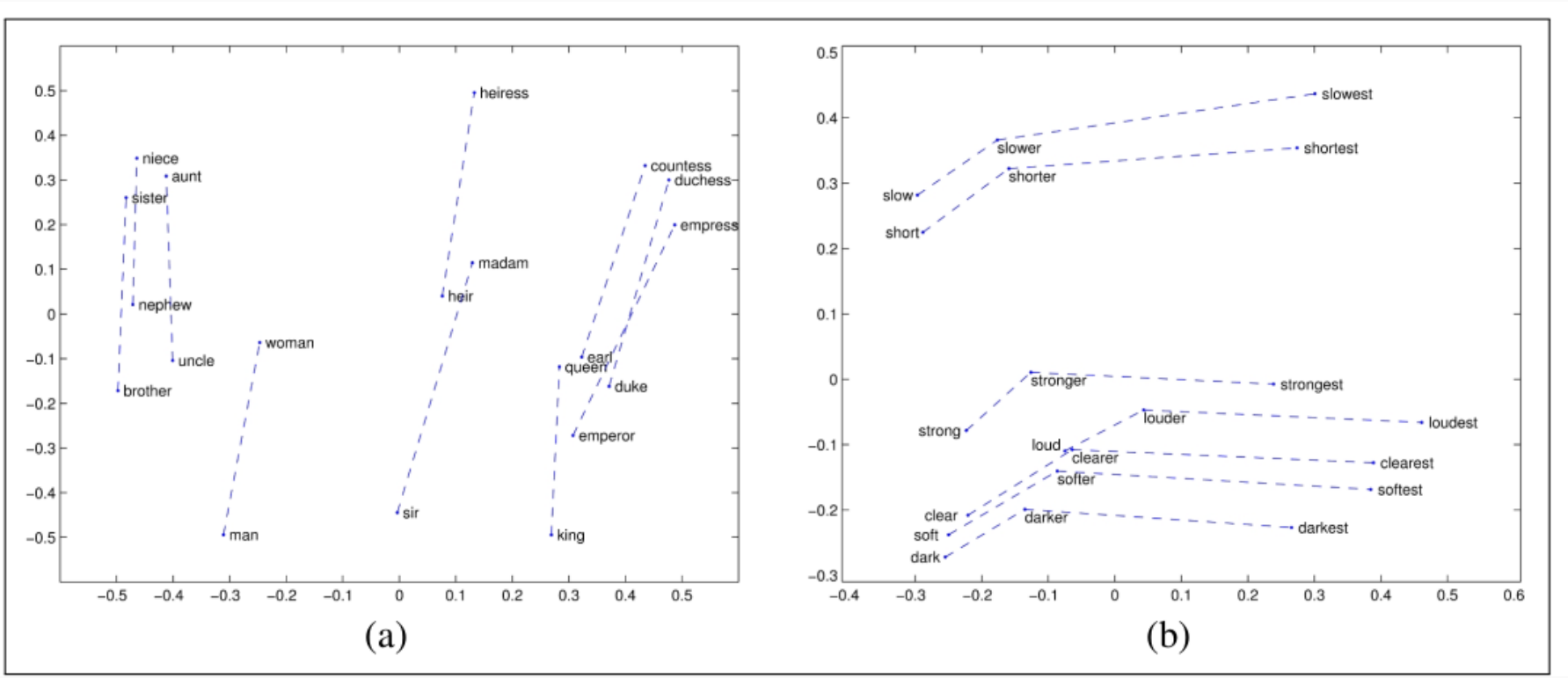
http://projector.tensorflow.org/



Analogy

- The semantic property of embeddings is their ability to capture relational meanings. Mikolov et al. (2013b) showed that the offsets between vector embeddings can capture some analogical relations between words.
- For example, the result of the expression
  - *vector(king) - vector(man) + vector(woman)* $\approx$ *vector(queen)*
  - *vector(Turkey) -vector(Ankara)* $\approx$ *vector(Canada)* - vector(Ottawa)

Vector Offset



(Dan Jurafsky and James H. Martin, 2018)

Detection of Change in the Senses of AI

The south-end of this dimension is populated by words such as “killer robots”, “humanity” and “Kurzweil” that represent more philosophical-speculative side of AI.

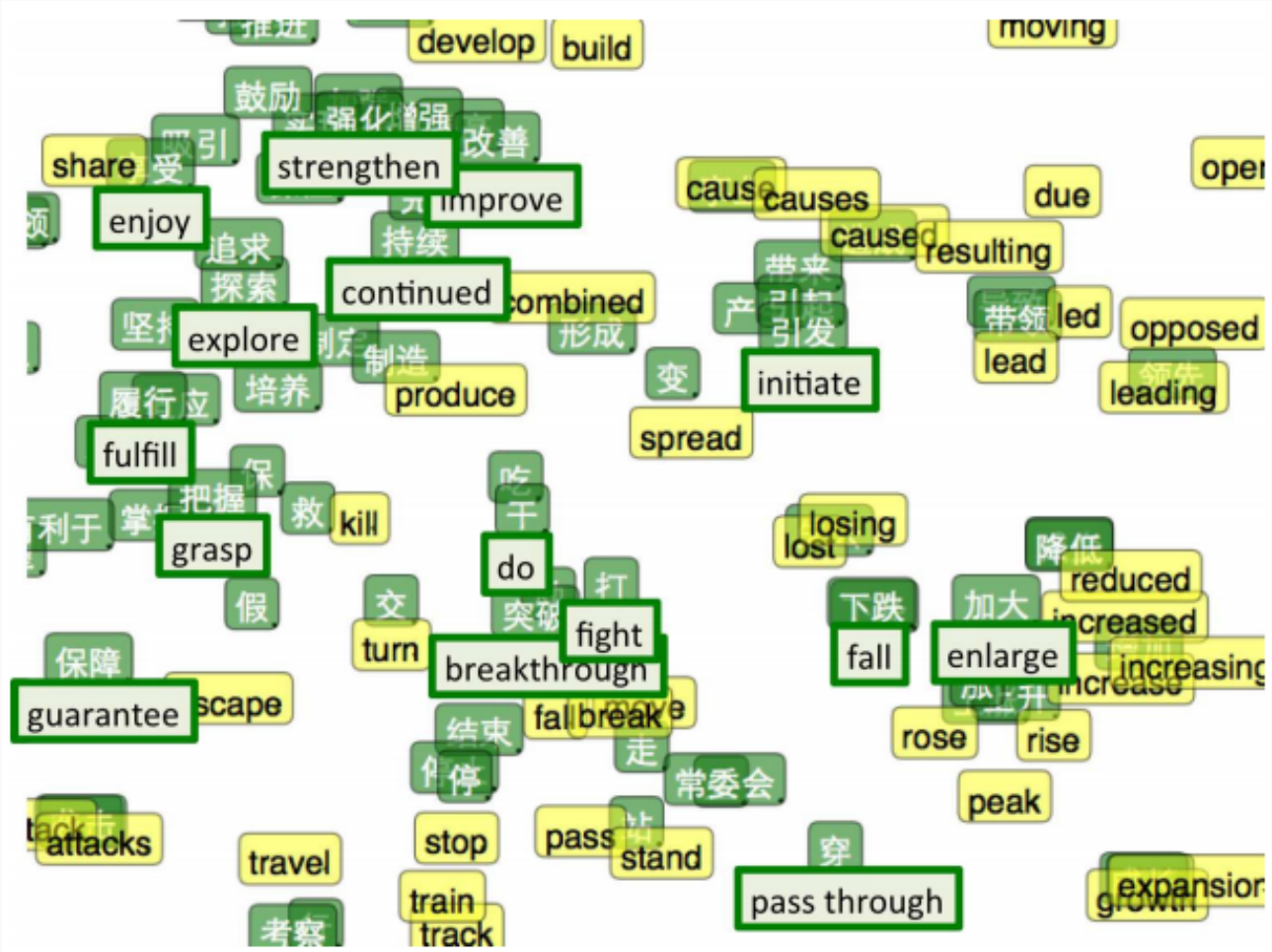
The north end has the words such as “IOT”, “ML” which represent more concrete app of AI. Hence, over the years, the sense of AI changes from more speculative to more concrete.

Detection of Change in the Senses of AI in Popular Discourse via Word Embeddings, Cicling 2018, Hanoi

nlp.stanford.edu, Word Embeddings for Historical Text

Bilingual

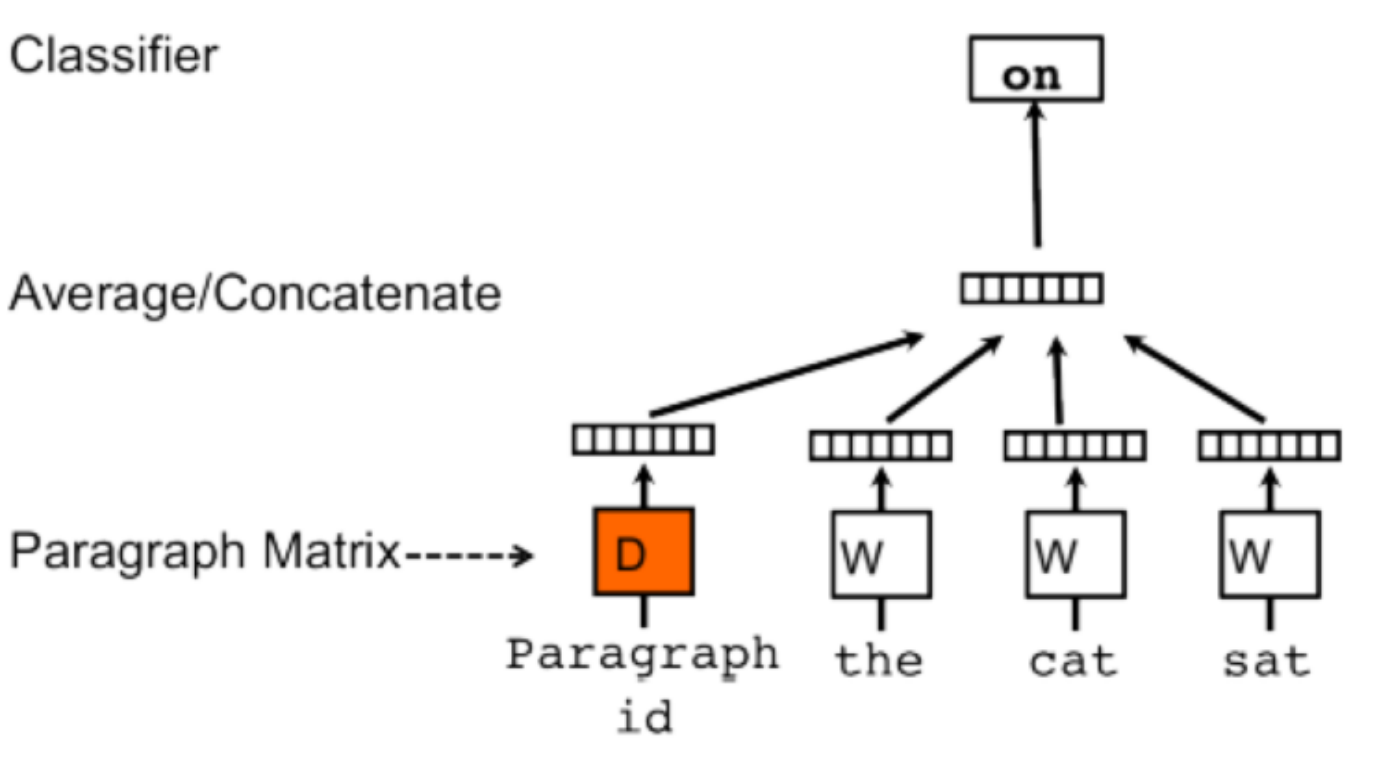
t-SNE visualization of bi-lingual word embeddings



(Socher et al.(2013a))

Alternative Document Representation: Paragraph Embeddings

Doc2Vec model (Mikolov et al., 2014)



Final Remarks for document representation

Feature Elimination

- TF-IDF(BOW) or PMI representaiton can be reduced by PCA or LSI
- n-gram also improves document representation success
- Stemming, noun phrase detection, stopword elimination, low freq and high freq word elimination, tokenization
- Post Tagging, NER
- multi-nominal NB, SVM, Logistic Reg have been found very successful especially for Document Classifications