HUST

ĐẠI HỌC BÁCH KHOA HÀ NỘI

HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY

ONE LOVE. ONE FUTURE.

ĐẠI HỌC
BÁCH KHOA HÀ NỘI
HANOI UNIVERSITY
OF SCIENCE AND TECHNOLOGY

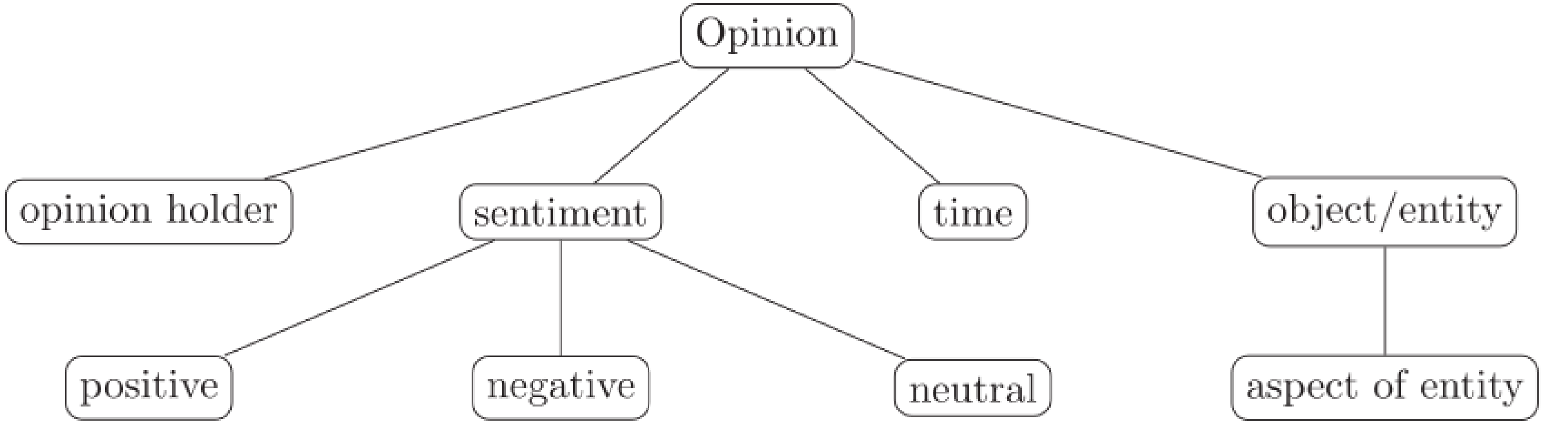
WEB MINING

LECTURE 05: OPINION MINING (1/3)

Content

1. Problems in opinion mining
2. Unsupervised Sentiment Analysis
3. Supervised Sentiment Analysis

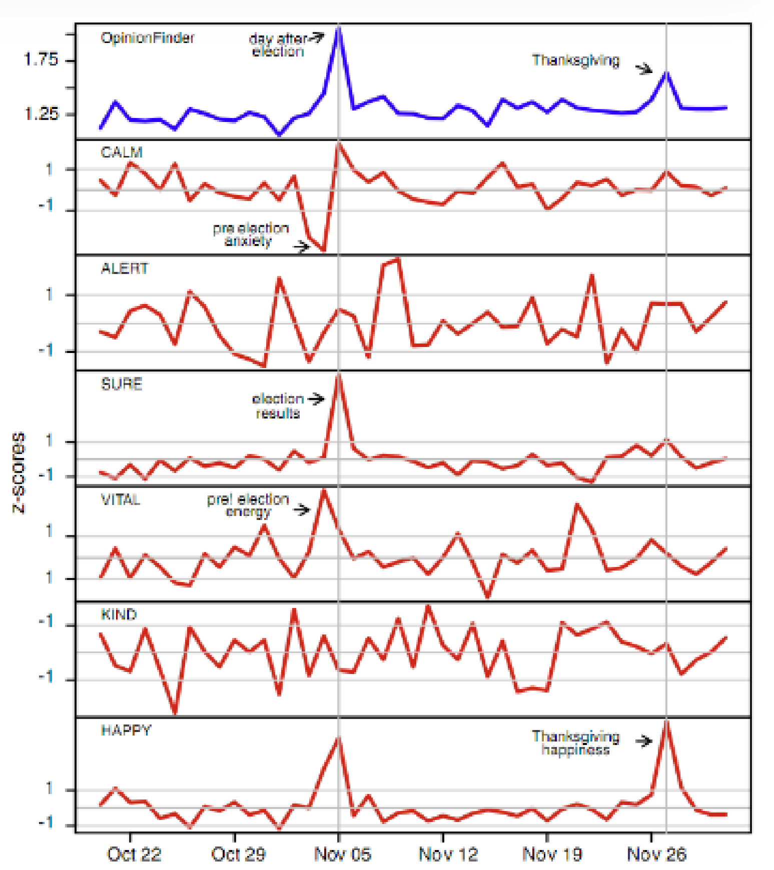
1. Problems in opinion mining



from M.D. Munezero et al. Are they different? Affect, feeling, emotion, sentiment, and opinion detection in text, IEEE Trans. Affect. Comput. 5 (2) (2014) 101–111

Applications

- Customer service
- Advertising, marketing
- Social credit, personal finance
- National security
- Social policies



Problem 1: Sentiment Analysis

Classify comments and reviews into one of three classes:

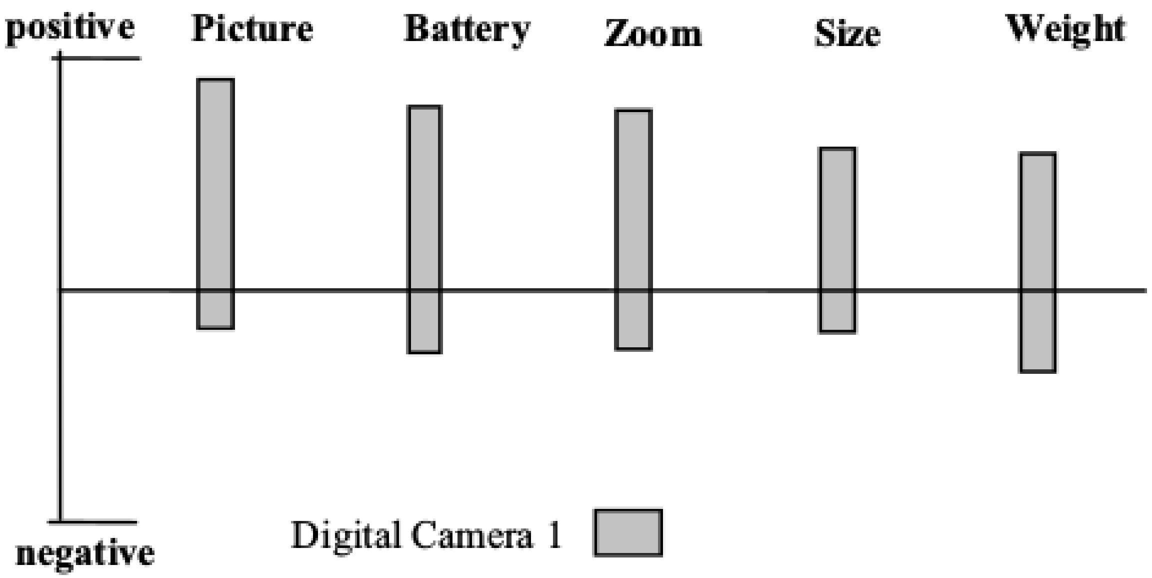
- Positive
- Negative
- Neutral

"BPhone 3, chất đến từng chi tiết"

Problem 2: Opinion summarization

Includes two sub-problems:

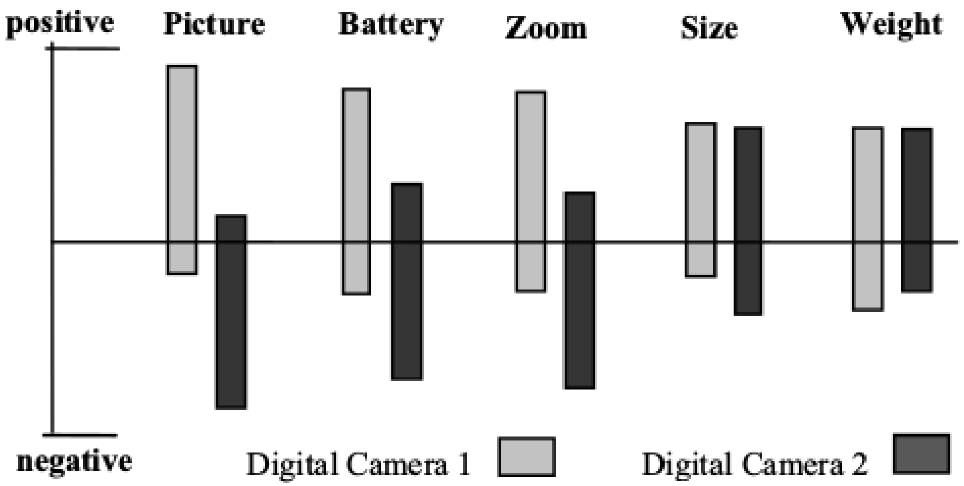
- Define aspect
- Categorize sentiment with each aspect



(A) Feature-based summary of opinions on a digital camera

Problem 3: Comparative opinions

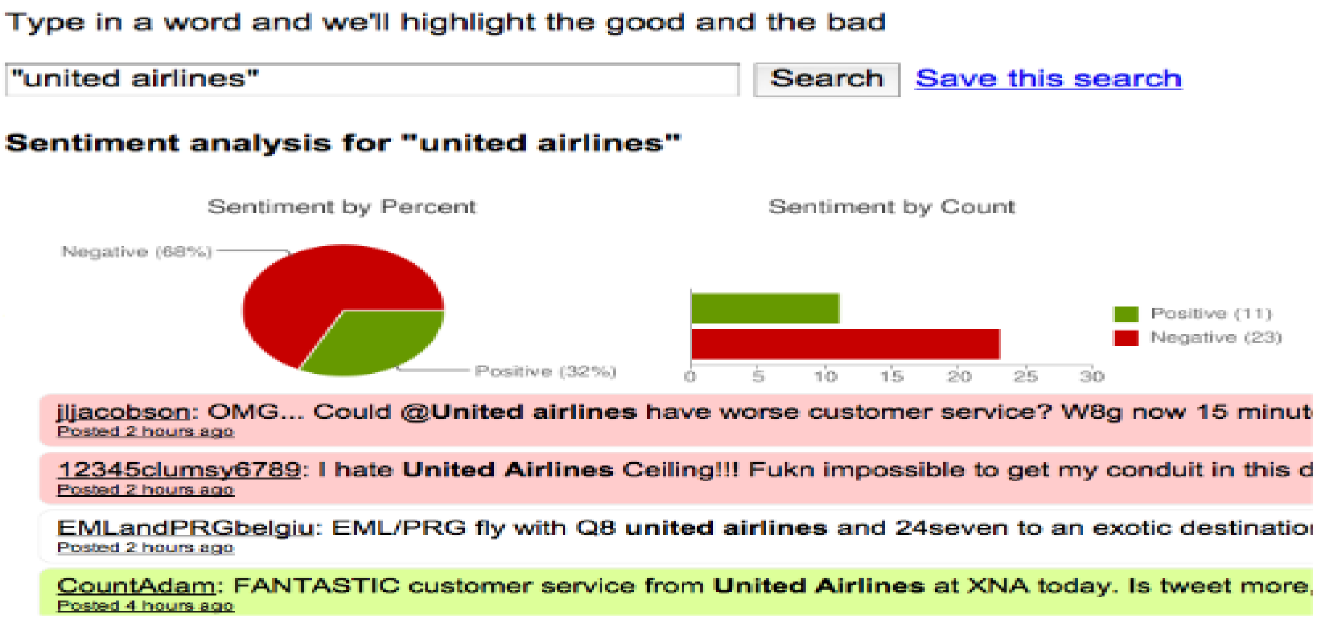
- Comparative opinions
  - Object A and object B
  - Object A and object B on aspect s
  - Object A with other objects



(B) Opinion comparison of two digital cameras

Problems 4: Opinions Searching

- Searching for Opinions on an object
- search host architecture based



Problem 5: Opinions Filtering

	Hype spam	Defaming spam
Good product	1	**2**
Bad product	**3**	4
Average product	**5**	**6**

	Example	Sentiment
Introverted sentiment	*Thật vinh dự và tự hào cho tôi khi được xem bóng đá Việt Nam chơi ở sân World Cup*	**Positive**
Extroverted sentiment	*Nur Farahain còn nổi tiếng là giáo viên thân thiện và hòa đồng với học sinh.*	**Positive**
Mood	*Thí sinh hồi hộp, gục trên bàn vì mệt mỏi*	**Negative**
Attitude	*Hết lòng vì nhà chồng nhưng tôi vẫn bị mẹ chồng ghét*	**Negative**
Character	*Em tự thấy mình khá năng động, biết đàn.*	**Positive**

Problem definition

- Requires **sentiment** recognition of a **subject** towards the **object** mentioned in the document.
- Simplify the problem given **subject** and **object** assumption

Document	**Sentiment**
Logitech pin trâu thôi rồi, mua 1 con B175 cùi mà cục pin theo chuột 3 năm chưa phải thay! ai chê thì chê chứ tôi thấy chuột Logitech xài hơi bị thích !	**Positive**
Hàng cùi bắp giá đắt. Lại còn nhái iphone để loa bên dưới nữa.	**Negative**
Đang dùng Logitech G502 mà nhìn thấy con này mà.......	**Neutral**

Methods of sentiment analysis

Methods	knowledge base request	Custom request by field	Training data request
Sentiment dictionary	✓	✓	✗
Unsupervised	✗	✗	✗
Supervised	✗	✓	✓

Sentiment Analysis based on dictionary

thực_sự là mình **rất sợ trà_sữa trân_châu** . **hầu_hết** các **cửa_hàng toàn nhập nguyên_liệu** từ **trung_quốc** với **giá rất rẻ** , vì mình có thằng bạn nó cũng **làm quán trà_sữa** nó **toàn lấy** từ **trung_quốc** . thế **mới** có **lãi cao** vì **thuê mặt_bằng rất đắt_đỏ rồi** . nên các bạn **hãy cân_nhắc** có **nên dùng trà_sữa ko nhé**

pos = 2

neg = 3

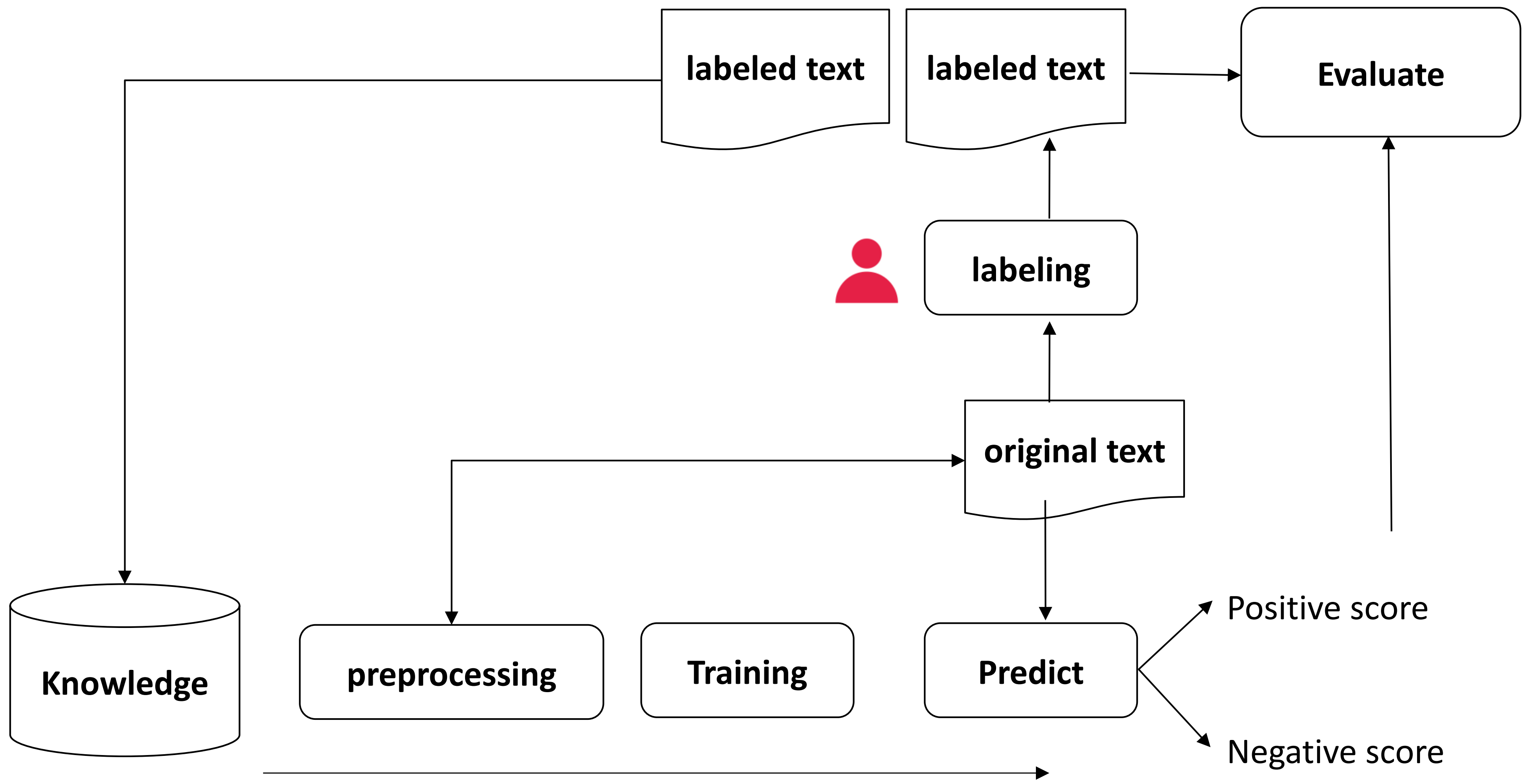
score = pos - neg = 2 – 3 = -1 < 0

Negative

Sentiment lexicon	
sợ	negative
rẻ	positive
lãi	positive
đắt đỏ	negative
cân nhắc	negative

https://github.com/stopwords/vietnamese-stopwords

Supervised Sentiement Analysis



2.2 Unsupervised Sentiment Analysis

- P. Turney. "*Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews*". ACL'02
- Algorithm:
  - **B1**. Extract opinion phrases
  - **B2**. Identify semantic/opinion orientation
  - **B3**. Determine the sentiment
- Apply to Vietnamese data

Step1. Extract opinion phrases

- Identify language patterns with potential for opinion expression:
  - *NN+JJ: commonnoun + adjective ('máy mới')*
  - *RB+JJ: adverb + adjective ('rất tốt')*
  - *RB+VA: adverb + verb adjective ('rất khỏe')*
  - *RB+VB: adverb + verb ('rất muốn')*
  - *VB+RB: verb + adverb ('chạy mượt')*
- Require document to be POS tagged

Step1. Extract opinion phrases (cont.)

first word	từ thứ hai
NN	JJ
RB	JJ/VA
RB	VB
VB	RB

Thực_sự là mình rất sợ trà_sữa trân_châu . Hầu_hết các cửa_hàng toàn nhập nguyên_liệu từ trung_quốc với giá rất rẻ , vì mình có thằng bạn nó cũng làm quán trà_sữa nó toàn lấy từ trung_quốc . Thế mới có lãi cao vì thuê mặt_bằng rất đắt_đỏ rồi . Nên các bạn hãy cân_nhắc có nên dùng trà_sữa ko nhé

Step1. Extract opinion phrases (cont.)

First word	Từ thứ hai
NN	JJ
RB	JJ/VA
RB	VB
VB	RB

Thực_sự là mình rất/RB sợ/VB trà_sữa trân_châu . Hầu_hết các cửa_hàng toàn nhập nguyên_liệu từ trung_quốc với giá rất/RB rẻ/VA , vì mình có thằng bạn nó cũng làm quán trà_sữa nó toàn lấy từ trung_quốc . Thế mới có lãi/NN cao/JJ vì thuê mặt_bằng rất/RB đắt_đỏ/VA rồi . Nên các bạn hãy cân_nhắc có nên dùng trà_sữa ko nhé

Step2. Identify Opinion Orientation

- For each extracted phrase t, necessary to determine opinion orientation of this phrase, SO(*t*)
- Assumption:
  - ‘*tốt*’ : positive
  - ‘*kém*’: negative
- SO(*t*) = sim(*t*, ‘*tốt*’) - sim(*t*, ‘*kém*’)

‘tốt’ *t* ‘kém’

Step2. Identify Opinion Orientation

- Determine the similarity of two phrases based on the likelihood of co-occurrence on a large corpus
  - Large Text Set: Web Text
  - Possibility co-occurrence: Pointwise Mutual Information (PMI)
  - SO(*t*) = PMI(*t*; ‘tốt’) - PMI(*t*; ‘kém’)

Step2. Identify Opinion Orientation

$$\mathrm{PMI}(t_1; t_2) = \frac{\Pr(t_1, t_2)}{\Pr(t_1)\Pr(t_2)} = \frac{\Pr(t_1 | t_2)}{\Pr(t_1)} = \frac{\Pr(t_2 | t_1)}{\Pr(t_2)}$$

$P(t_1)$: Probability occurrence t_1 in corpus
$P(t_1 | t_2)$: Probability occurrence t_1 when t_2 2occurrenced

$P(t_1 | t_2) = (\mathrm{count}(t_1, t_2) + 1) / (\mathrm{count}(t_2) + V)$

$P(t_1) = (\mathrm{count}(t_1) + 1) / (\mathrm{sum}_{t'}\ \mathrm{count}(t') + V)$

V: Vocab size

Step3. Determine the sentiment

- Assume document *d* consists a set of opinion phrases *T* extracted from step 2
- For each $t \in T$, calculate SO(t)
- Opinion orientation

SO(d) = $\text{sum}_{t \in T}$SO(t)

- SO(d) > 0: Positive document
- SO(d) < 0: Negative document

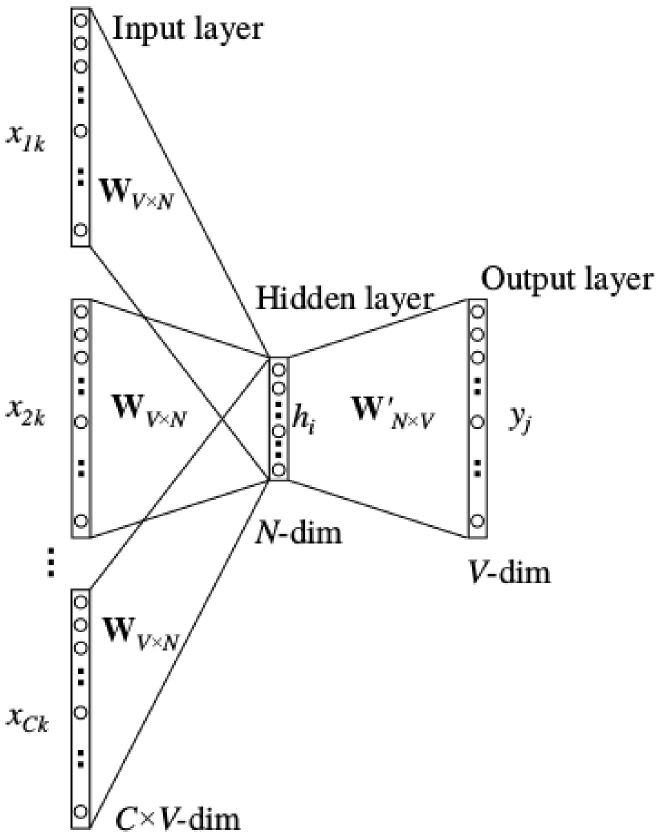
3. Supervised Sentiment Analysis

- Yoon Kim. "*Convolutional Neural Networks for Sentence Classification*". EMNLP 2014
- Using CNN model to classify reviews
- Using pre-trained word embedding on a large data set as a word representation vector
- Concatenate the ordered word representation in the text to serve as a 2D input signal for the CNN

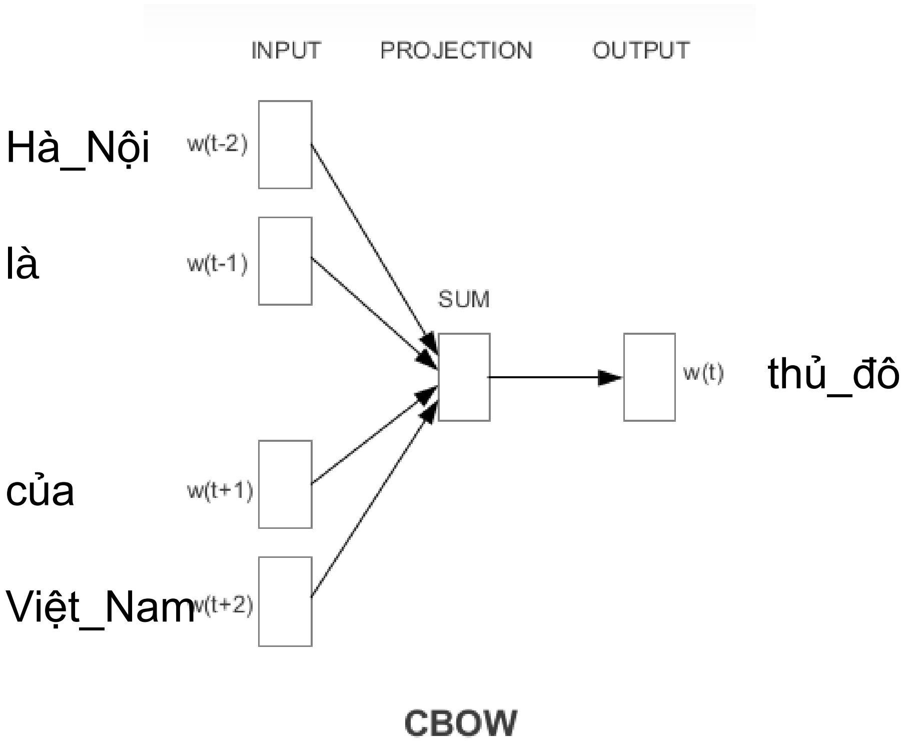
Word2vec

- Using a neural network to learn a language model task:
  - CBOW: Use surrounding words in a window to predict the center word
  - Skip-gram: Use focus words to predict surrounding words
- Leverage large amounts of learning data without labeling (!)
- Generates a vector representation of a word that exhibits some semantic relations.

CBOW



Input layer
x_{1k}
$W_{V\times N}$
x_{2k}
$W_{V\times N}$
x_{Ck}
$W_{V\times N}$
$C\times V$-dim
Hidden layer
h_i
N-dim
$W'_{N\times V}$
Output layer
y_j
V-dim



INPUT
PROJECTION
OUTPUT
Hà_Nội
w(t-2)
là
w(t-1)
của
w(t+1)
Việt_Nam
w(t+2)
SUM
w(t)
thủ_đô
CBOW

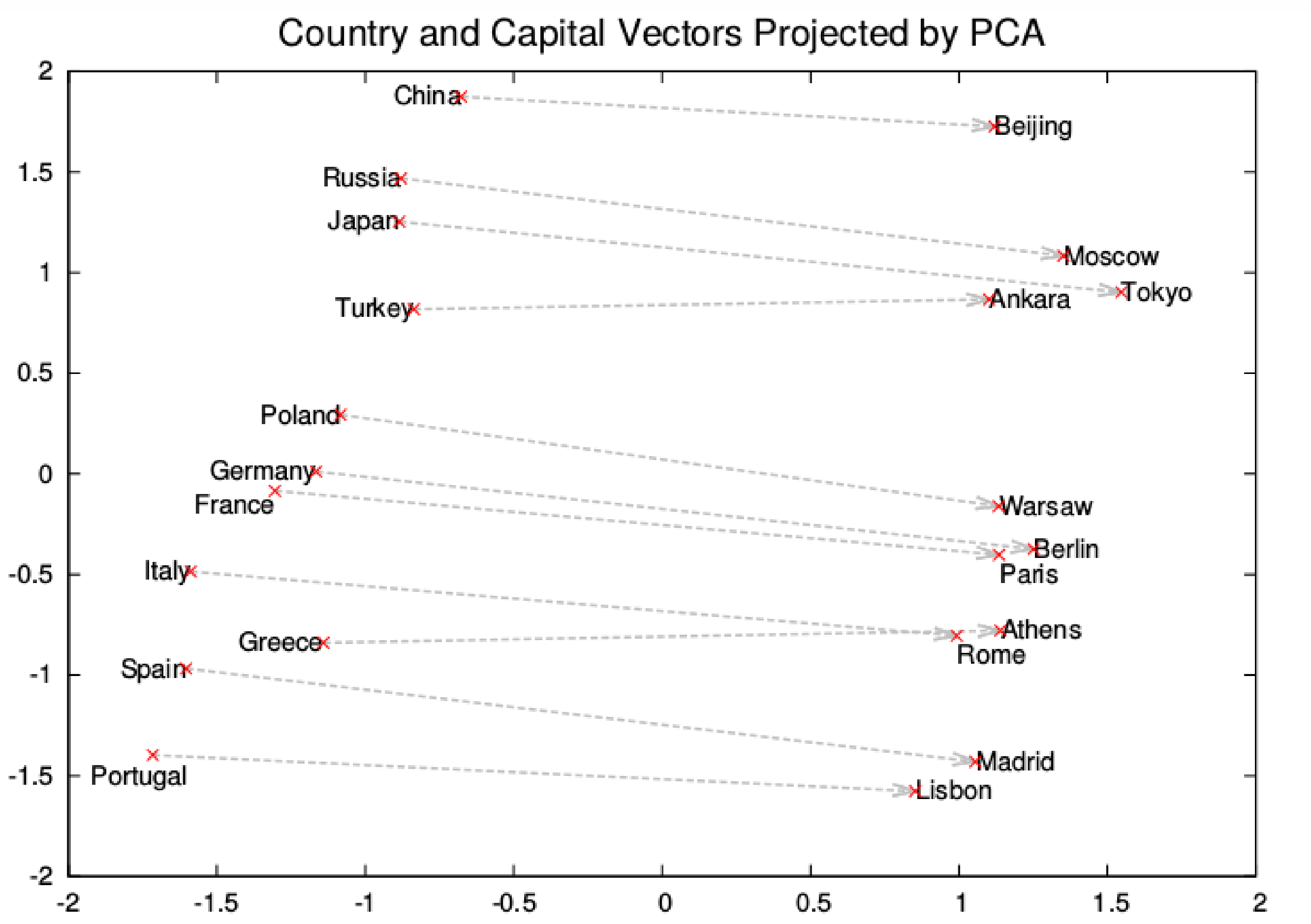
CBOW (cont.)

- The input layer consists of V neurons that represent words in the context in the form: one-hot
- The hidden layer consists of n neurons
- The output layer consists of V neurons used to predict center word
- The weight between the input layer and the hidden layer after learning is used as a lookup table of word representation

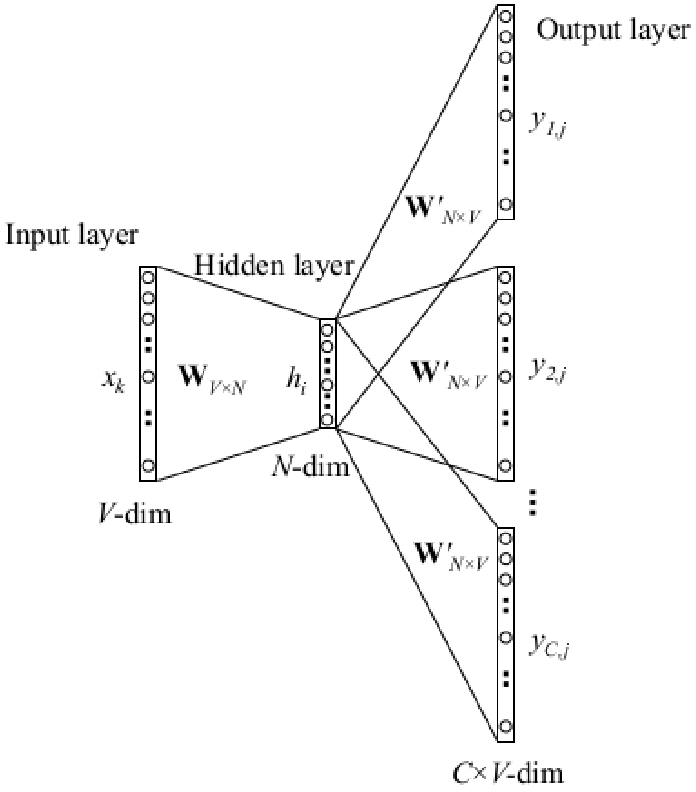
king – queen = man - …

- Vector representation ‘king’: ***a***
- Vector representation ‘queen’: ***b***
- Vector representation ‘man’: ***c***
- Calculate vector ***d*** = ***a*** – ***b*** + ***c***
- search word ***d'*** whose distance (Euclidean, cosine) to **d** is closest: ***d'*** ~ 'woman'

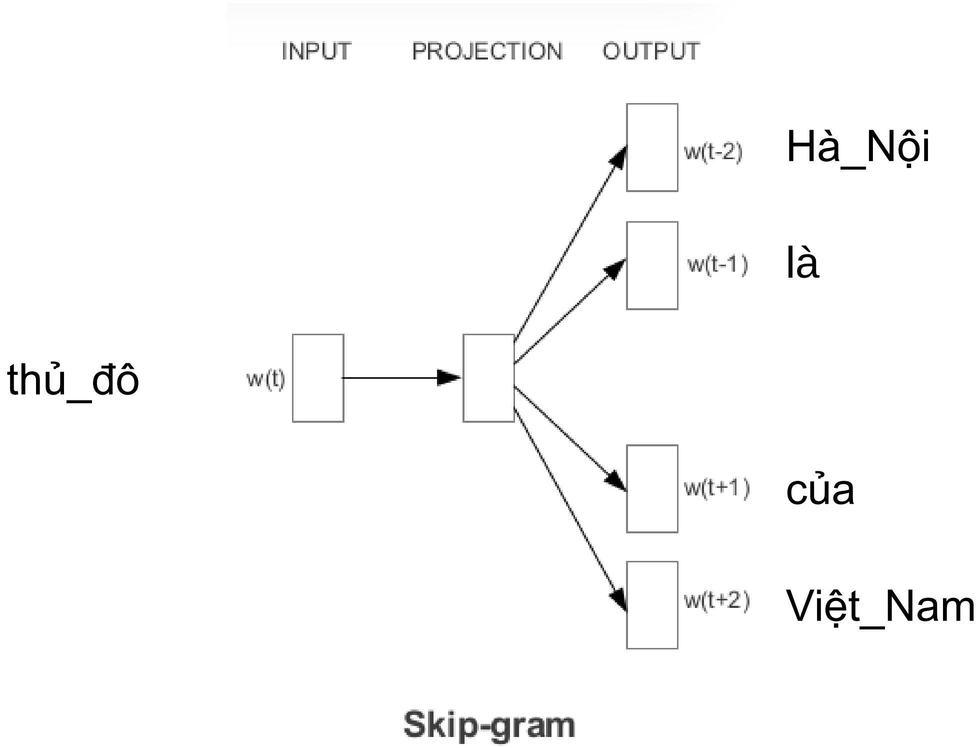
Visualization word representation



Skip-gram

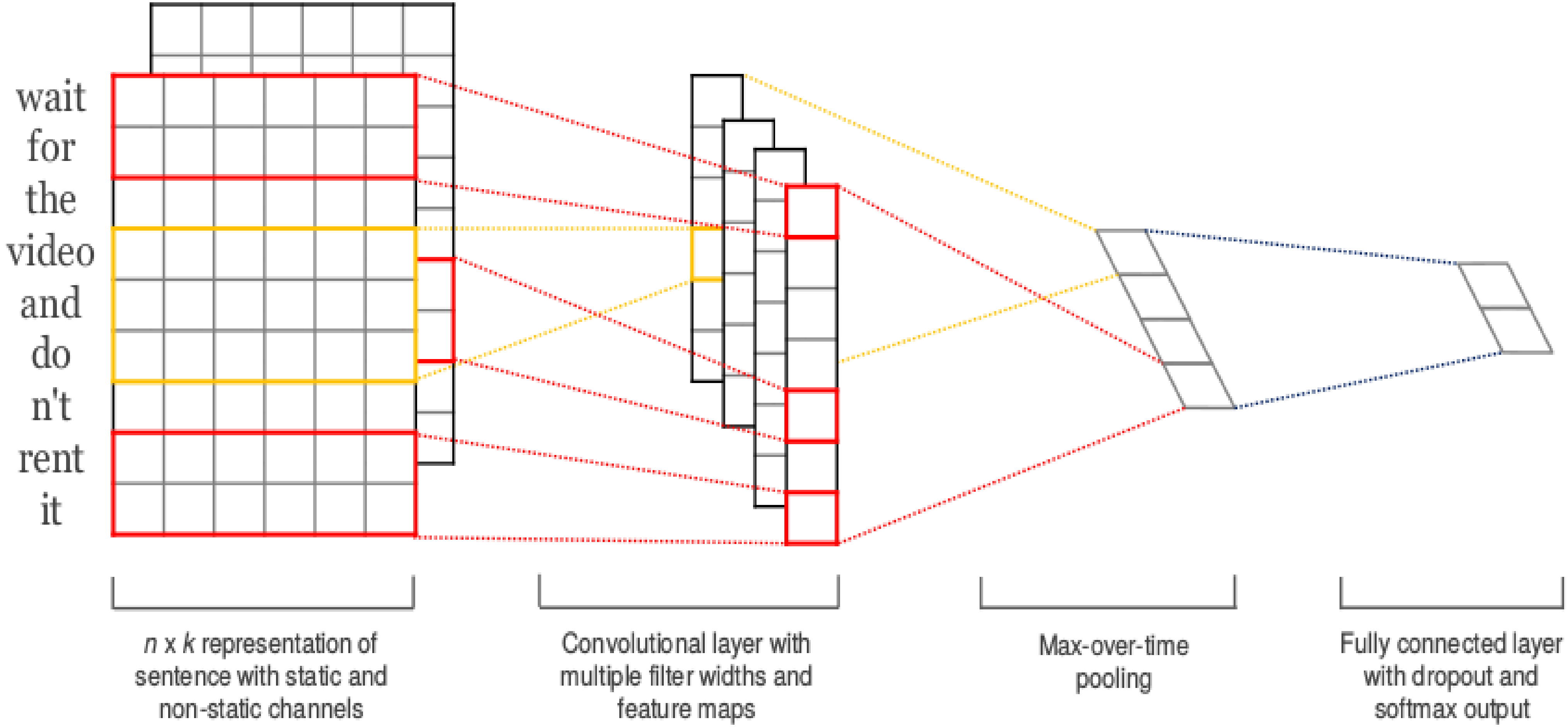


Output layer
Input layer
Hidden layer
$y_{1,j}$
$\mathbf{W}'_{N\times V}$
x_k
$\mathbf{W}_{V\times N}$
h_i
$y_{2,j}$
$\mathbf{W}'_{N\times V}$
N-dim
V-dim
$\mathbf{W}'_{N\times V}$
$y_{C,j}$
$C\times V$-dim



INPUT
PROJECTION
OUTPUT
thủ_đô
w(t)
w(t-2)
Hà_Nội
w(t-1)
là
w(t+1)
của
w(t+2)
Việt_Nam
Skip-gram

Model architecture



Input layer

- $\boldsymbol{x}_i \in R^k$ is continuous representation of word *i*
  - Randomly initialized and weights updated during learning
  - Initialized based on a pre-trained representation of a large corpus
    - Updated during training
    - “Freeze” in training
- Input consists words $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n$ in order
- Document representation is a concatenation of word representations in the order appear in document

$$\mathbf{x}_{1:n} = \mathbf{x}_1 \oplus \mathbf{x}_2 \oplus \ldots \oplus \mathbf{x}_n ,$$

Convolutional layer

- Each filter $\boldsymbol{w} \in \mathrm{R}^{hk}$ scans a window of h consecutive words
- $\boldsymbol{x}_{i:i+h}$ to generate a feature c_i
  - Window width: h
  - Window height = word embedding dimension
- Each filter generates a feature map $\boldsymbol{c} \in \mathrm{R}^{n-h+1}$, $\boldsymbol{c} = [c_1, c_2, \ldots, c_{n-h+1}]$

$$c_i = f(\mathbf{w} \cdot \mathbf{x}_{i:i+h-1} + b).$$

Pooling layer

- For each feature map c, apply max pooling to get the maximum value
- Apply windows h ∈ [3, 4, 5]
- For each value of h there are 100 filters
- Total number neurons in the pooling layer: 100 x 3 = 300

Fully connected layer

- Adjustment technique: Apply dropout at pooling layer with dropout ratio $p = 0.5$
- Number neurons in output layer:
  - 2: only positive and negative labels
  - 3: positive, neutral, negative

Datasets

- **MR**: Movie commentary with each comment being a sentence. Label positive/negative
- **SST-1**: Extension MR set with 5 labels (very positive, positive, neutral, negative and very negative)
- **SST-2**: Similar to SST-1 but removes neutral label and has only two positive and negative labels
- **CR**: Product reviews. Label positive/negative.

Models

- **CNN-rand**: Embedded words are randomly initialized and updated during training
- **CNN-static**: Using pretrained word2vec embedding, word representations (including randomly initialized OOV words) are kept weighted
- **CNN-non-static**: The initial word representation in word2vec is fine-tuned during the training
- **CNN-multichannel**: hybrid static and non-static

Experimental results

Model	MR	SST-1	SST-2	Subj	TREC	CR	MPQA
CNN-rand	76.1	45.0	82.7	89.6	91.2	79.8	83.4
CNN-static	81.0	45.5	86.8	93.0	92.8	84.7	**89.6**
CNN-non-static	**81.5**	48.0	87.2	93.4	93.6	84.3	89.5
CNN-multichannel	81.1	47.4	**88.1**	93.2	92.2	**85.0**	89.4
RAE (Socher et al., 2011)	77.7	43.2	82.4	–	–	–	86.4
MV-RNN (Socher et al., 2012)	79.0	44.4	82.9	–	–	–	–
RNTN (Socher et al., 2013)	–	45.7	85.4	–	–	–	–
DCNN (Kalchbrenner et al., 2014)	–	48.5	86.8	–	93.0	–	–
Paragraph-Vec (Le and Mikolov, 2014)	–	**48.7**	87.8	–	–	–	–
CCAE (Hermann and Blunsom, 2013)	77.8	–	–	–	–	–	87.2
Sent-Parser (Dong et al., 2014)	79.5	–	–	–	–	–	86.3
NBSVM (Wang and Manning, 2012)	79.4	–	–	93.2	–	81.8	86.3
MNB (Wang and Manning, 2012)	79.0	–	–	**93.6**	–	80.0	86.3
G-Dropout (Wang and Manning, 2013)	79.0	–	–	93.4	–	82.1	86.1
F-Dropout (Wang and Manning, 2013)	79.1	–	–	**93.6**	–	81.9	86.3
Tree-CRF (Nakagawa et al., 2010)	77.3	–	–	–	–	81.4	86.1
CRF-PR (Yang and Cardie, 2014)	–	–	–	–	–	82.7	–
SVM_S (Silva et al., 2011)	–	–	–	–	**95.0**	–	–

Fine-tuned word embedding

	Most Similar Words for	
	Static Channel	Non-static Channel
bad	*good*	*terrible*
	terrible	*horrible*
	horrible	*lousy*
	lousy	*stupid*
good	*great*	*nice*
	bad	*decent*
	terrific	*solid*
	decent	*terrific*
n't	*os*	*not*
	ca	*never*
	ireland	*nothing*
	wo	*neither*

!	*2,500*	*2,500*
	entire	*lush*
	jez	*beautiful*
	changer	*terrific*
,	*decasia*	*but*
	abysmally	*dragon*
	demise	*a*
	valiant	*and*

HUST

hust.edu.vn   fb.com/dhbkhn

THANK YOU !