워드 임베딩과 문서 표현

- 워드 임베딩이란?
- 워드 임베딩 방법론: word2vec, fastText, GloVe
- 문서표현 방법: ELMo, BERT

문서 분류와 문서 클러스터링 문제

- 머신러닝 방법론
  - 문서 분류: kNN, NB, SVM, random forest 등
  - 문서 클러스터링: K-Means, DBSCAN
- 주요 이슈
  - 문서 벡터를 어떻게 구성할 것인가?
  - 유사도 계산 기법은?

문서 벡터의 구성

- 문서의 특징을 가장 잘 표현하는 feature vector
- Keyword 추출 및 키워드 가중치 계산: TF-IDF

- 예) SVM 학습데이터: example1.tar.gz -- 로이터 뉴스기사
    - train.dat(2천개), test.dat(600개)
    - words: 9,947 terms are extracted

  https://download.joachims.org/svm_light/examples/example1.tar.gz

SVM example1: train.dat, test.dat, words

train.dat | test.dat | words

```
977 1 6:0.0394818835476958 11:0.010436379800545 15:0.00795697050873869 26:0.0301567899808728 27:0.0135249148075739 29:0.0674778841926948 31:0.0177039764265539 3
978 1 6:0.0221992617185068 15:0.0380283377329407 41:0.0255548733073093 63:0.0119959159683644 67:0.0313931869855986 134:0.0397475335703676 142:0.0028782959386757
979 1 6:0.00504987568654403 11:0.0680773866292088 15:0.0346026603140889 26:0.0655717624393253 28:0.0372907813132578 29:0.0550204920060878 63:0.0218305942439399
980 1 15:0.188437715840106 28:0.203076572401634 158:0.197385390277138 176:0.142546636568 204:0.165090087892139 2216:0.37464991325394 7038:0.570477684779504 8747
981 1 6:0.0116994617445259 15:0.0400834125166914 26:0.151915487389957 27:0.102198080077658 31:0.044592083727845 63:0.0252883653054868 64:0.202944862888027 75:0.
982 1 6:0.0120259484621456 27:0.0233444518081582 28:0.0296018517433357 29:0.0291172658749587 31:0.0305576508527073 42:0.0239832788068884 80:0.0261007140363266 88:
983 1 6:0.0104795842764911 15:0.0718080042878157 29:0.0380598070594226 63:0.0226516023745742 75:0.0212386572738854 81:0.0638057422866432 142:0.00271750882930389
984 1 6:0.0288297267873363 15:0.0493866237949364 29:0.0523519736161364 67:0.0815393786743864 81:0.175531974270551 189:0.107869141401183 365:0.152731612076903 539
985 1 6:0.0176472501759912 15:0.0151152682071138 26:0.0572866228831546 27:0.0128461400334668 29:0.128182744488119 31:0.0336309342514079 41:0.0203147856081126 47
986 1 6:0.0354004780447618 11:0.0198847471725369 12:0.0596605697051269 17:0.0241126491435809 28:0.0980304937507854 29:0.0642838174037799 31:0.0168659584031329 3
987 1 6:0.021155606005887 15:0.0483206778278397 26:0.0915672950771417 27:0.0205333502979564 29:0.0256110122172014 33:0.0390227465462309 60:0.107048834814379 75:
988 1 6:0.00849877661180451 15:0.0291175761976879 17:0.0463107912454728 52:0.0692892402227694 59:0.0527728839100392 94:0.0510197567642055 97:0.0530310748091109
989 1 6:0.0150788713695264 15:0.0258307875434376 28:0.111349848906196 29:0.0273817605668505 31:0.0287362918617146 205:0.0596573795865003 365:0.0798835295789082
990 1 6:0.0210434060232993 15:0.0480644063124127 26:0.0607211084137654 29:0.0254751827295459 63:0.0151617612966066 67:0.119034513576201 88:0.0535618642662177 142
991 1 6:0.0232948343287734 12:0.10469034423651 17:0.0211560294493696 29:0.0564015312466947 31:0.0443937176554712 33:0.0214843388460535 37:0.0183698241040129 39:
992 1 6:0.0112844076408572 11:0.101416751557565 15:0.0128871332246437 26:0.0325613514745168 29:0.0409827686155792 31:0.0430101197241555 32:0.0281972047912574 60
993 1 6:0.0224033134947043 15:0.00959447190988464 26:0.0484838585315036 28:0.0206796443140544 29:0.0101705583757386 31:0.0320210382201272 41:0.0644744232128918
994 1 6:0.0279060938200725 15:0.0318695992102037 26:0.0402617557817681 31:0.0354543624688264 63:0.0402126473953027 68:0.228419603673353 75:0.0188521461303572 81
995 1 68:0.198886860352125 111:0.20260601890649 232:0.287562034965263 339:0.251990753968374 829:0.24033777668258 1074:0.42340552984138 2250:0.540359504256376 26
996 1 3:0.177784378824684 63:0.150761492639695 67:0.197270626984526 142:0.0271302453967192 441:0.24591625869489 553:0.445811082678889 979:0.534547130633568 1983
997 1 111:0.147621755758767 142:0.00612118404400153 196:0.219506078622476 229:0.123617517702769 321:0.313421052834445 388:0.245468280562 413:0.204100912026106 7
998 1 6:0.0161664810467732 17:0.0176185953138505 26:0.0699729868270837 27:0.0188291662262737 31:0.01232359389114 41:0.0148881482713775 48:0.0197801391509356 63:
999 1 6:0.0266122935137017 12:0.0448498065528929 15:0.0227940302344268 26:0.0287963357344921 27:0.0387443081136612 29:0.0241626654697638 31:0.025357953350038 63
1000 1 6:0.0228727734074645 29:0.0415347269430743 31:0.0435893825350942 80:0.148926894158521 142:0.00593124326743463 158:0.08208518996497 159:0.0688008021473813
1001 1 6:0.0122117884664382 26:0.158567961269673 27:0.0355577992343225 28:0.270533669388619 29:0.0443508349778489 31:0.0930895977358842 63:0.026395758584117 67:0
1002 -1 6:0.0131555009596025 9:0.100637011736296 27:0.0383056636817852 41:0.121152555280363 63:0.0568711828471545 142:0.0204684779863839 206:0.125474292043272 28
1003 -1 6:0.0199371587356936 27:0.0193507415553135 31:0.0253299246025387 37:0.0314440612832433 42:0.119281677802051 83:0.0344389590714789 127:0.0660942623191568
1004 -1 6:0.0309207391039684 11:0.0154386133478623 15:0.0117708049584728 18:0.183908724695778 27:0.0100037529424719 31:0.0392844336819212 33:0.0190117009606583 4
1005 -1 2:0.0388121013491587 6:0.028522387239428 29:0.0345292499704082 31:0.0362373559761895 41:0.175113569467451 42:0.113764060698306 84:0.0557988510673976 105:
1006 -1 8:0.114796369082114 10:0.0961912968972383 14:0.347110326909765 16:0.125504526122891 22:0.121619693949677 47:0.0539783388731893 52:0.092979060045452 70:0.
1007 -1 6:0.00965958410257285 8:0.388929428071602 9:0.443363585324567 13:0.256797994485874 33:0.0534529952115495 71:0.553256623583221 75:0.0587303629848309 106:0
1008 -1 8:0.274066364016247 9:0.208282937720586 71:0.389862582384596 90:0.380427615245882 119:0.284874303661294 142:0.0282416621354932 175:0.263558974417106 471:
1009 -1 6:0.0168880564119203 8:0.254990097857046 9:0.581356492198111 10:0.0712212659829517 14:0.171336615790234 71:0.241817581640326 75:0.0171132572004524 83:0.0
1010 -1 6:0.0236460984619881 9:0.271332049126534 13:0.157156628017677 58:0.0396549680936414 71:0.338584986089396 75:0.0359421257452925 84:0.0346944607580457 90:0
1011 -1 6:0.0173367283847702 11:0.155810985510061 50:0.493679308125092 51:0.117745623284991 74:0.14212355847612 75:0.0702716487028172 114:0.365205092684249 137:0
1012 -1 17:0.22807578599311 53:0.322318987913691 63:0.0904635467068251 101:0.813538792798633 106:0.292954447507752 114:0.293878057803022 142:0.0434115843710704
1013 -1 1:0.154152837554146 59:0.224129960648084 75:0.146304774421378 101:0.701624962203208 142:0.0187198517840518 143:0.318050438827384 264:0.297135048593728 30
1014 -1 11:0.022286148588098 13:0.0329615249807139 17:0.0540492747535355 34:0.0395243887745897 50:0.235375154063079 63:0.139358131649027 75:0.251279522973765 82:
1015 -1 6:0.0323448239501611 11:0.036336720977379 17:0.0220313237164471 18:0.151498321280615 25:0.0303439800602742 27:0.176588193774329 33:0.0223732163487488 36:
1016 -1 6:0.0412075353048041 17:0.269453402870798 27:0.0119986459914242 33:0.0456058183031041 34:0.0985210356685215 37:0.038994491119818 53:0.285618480059592 58:
1017 -1 11:0.0627037142231267 17:0.0760358001005813 50:0.264898106323813 63:0.0904833065364143 75:0.169678409920521 82:0.10466352249221 83:0.0361551644961694 101
1018 -1 6:0.0426936777639266 11:0.0319751986447939 12:0.0479678859135231 15:0.0243787326151545 25:0.213614165285845 27:0.0414379167768029 34:0.05670787742376 41:
1019 -1 6:0.0413908670825799 13:0.0250083505007003 17:0.0205039543549024 18:0.0402843917578667 27:0.0876511113325552 37:0.0178036259515365 42:0.0225124267009695
1020 -1 6:0.00844540399713867 9:0.129211380666948 17:0.046019957855102 26:0.0365540367709853 27:0.0245909909599499 31:0.0321893579704289 34:0.201916941229962 53:
1021 -1 6:0.0237557394349587 17:0.0431492335154117 37:0.187332843291327 41:0.0729243364672465 94:0.0950734521830662 142:0.00205340307267603 186:0.044935090217325
1022 -1 6:0.00986378931576122 10:0.0831962595721511 34:0.0786094688602254 48:0.0603431027002044 53:0.0759645452425385 72:0.0613846847948081 80:0.0642241096547967
1023 -1 6:0.0151837532833751 10:0.0640337825051005 17:0.0827379822705852 42:0.0454213057884445 72:0.0472460369673956 96:0.0470375194496628 109:0.0835473664290476
1024 -1 6:0.0044914051194482 9:0.274866972291733 13:0.0895522893261015 21:0.0543409137829537 69:0.0911344839855 98:0.0291343181339084 105:0.0954219449645924 106:
1025 -1 6:0.00730184678590241 18:0.019543290883565 22:0.0389341645599805 27:0.0106306133113289 42:0.0218430456220586 57:0.054513890242402 65:0.0261194144950109 8
1026 -1 6:0.0398887324888581 9:0.152570504610737 10:0.0841105085457473 17:0.054339549317375 22:0.106345320598117 27:0.0580732112027858 41:0.0918365230400542 42:0
```

train.dat | test.dat | words

```
1 v
2 ct
3 shr
4 net
5 qtr
6 said
7 rev
8 wheat
9 tonn
10 agricultur
11 bank
12 oil
13 export
14 grain
15 inc
16 corn
17 trade
18 rate
19 soybean
20 loss
21 olon
22 usda
23 th
24 div
25 dollar
26 share
27 pct
28 acquir
29 company
30 profit
31 corp
32 dividend
33 issu
34 import
35 crop
36 bond
37 market
38 money
39 barrel
40 debt
41 price
42 product
43 coupon
44 avg
45 qtly
46 curenc
47 note
48 record
49 underwrit
50 deficit
```

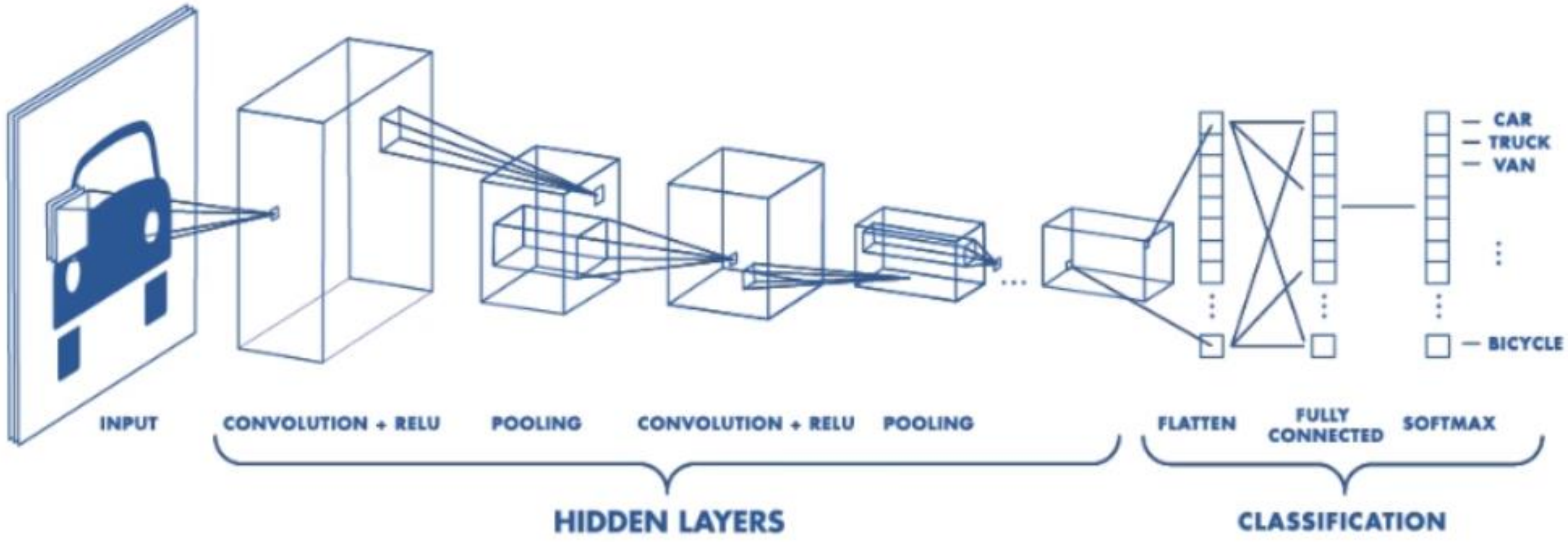
문서 벡터 구성의 문제점

- 문서 벡터의 차원수 문제
  - Feature(키워드) 개수: 수십만~수백만개
  - 저장 공간의 문제

- 차원 축소(dimensionality reduction)
  - Feature selection
  - 고차원 데이터를 저차원 공간으로 투영
    - PCA, LDA, LSA 등

심층 뉴럴네트워크를 이용한 차원 축소

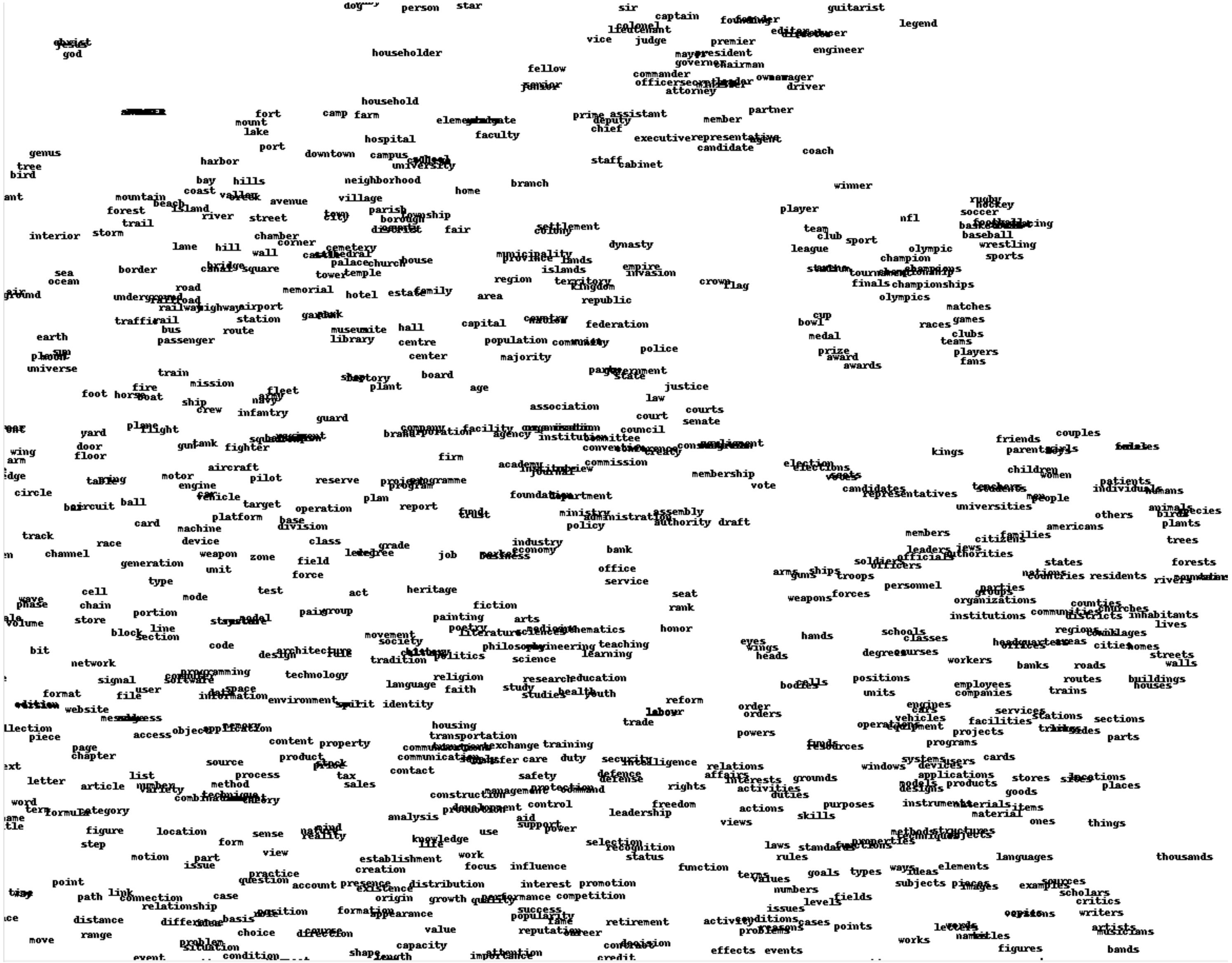
- 입력 벡터 → 은닉층 → 출력 벡터
- CNN(Convolutional Neural Network)

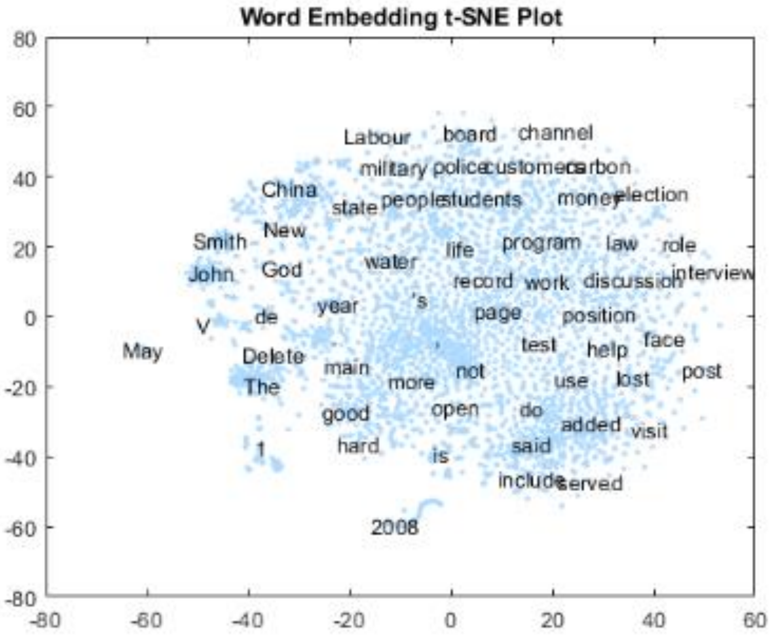


워드 임베딩: 단어 벡터(word vector) 구성

- 단어의 특징을 가장 잘 표현하는 벡터
  - “You shall know a word by the company it keeps”, J. R. Firth (1957)
  - “Efficient Estimation of Word Representations in Vector Space”
- 대규모 텍스트 말뭉치로부터 단어 벡터 학습
- DNN(Deep Neural Network)으로 구현

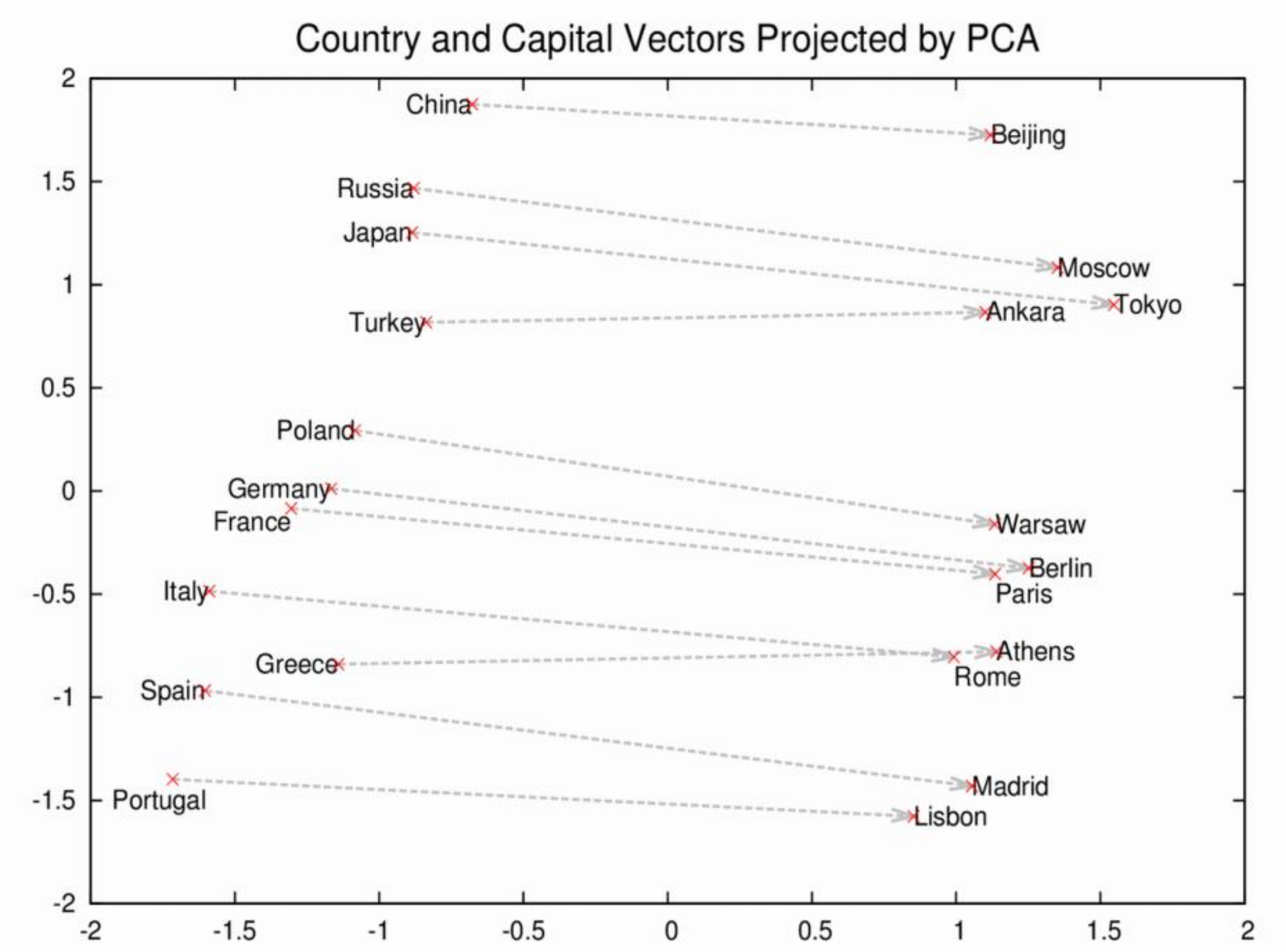
Joseph Turian's map of 2500 English words
produced by using t-SNE on the word feature vectors learned by Collobert & Weston, ICML 2008





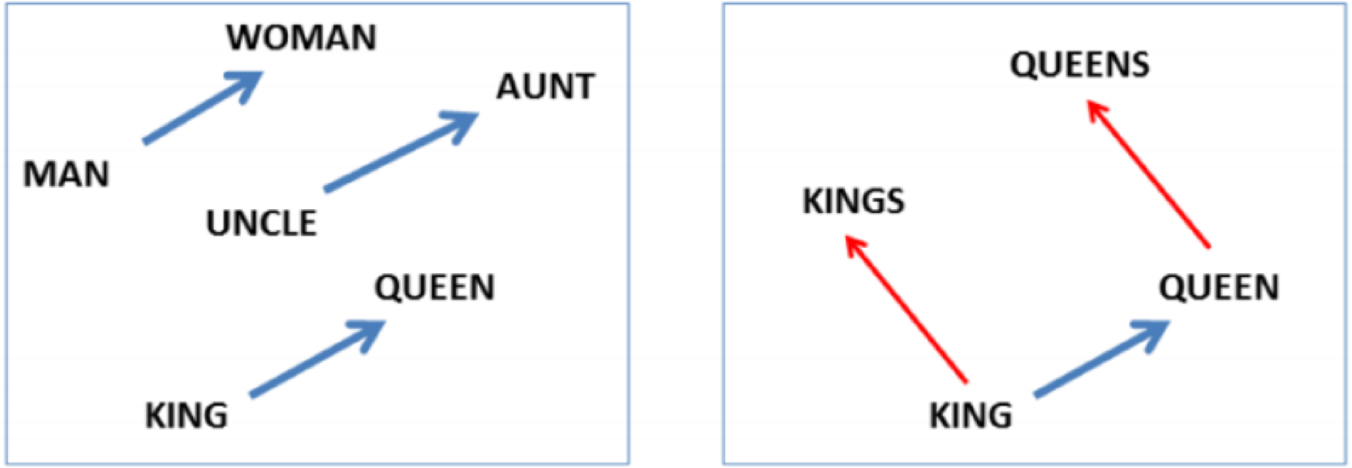
http://www.cs.toronto.edu/~hinton/turian.png

단어 벡터의 방향성



Country and Capital Vectors Projected by PCA
China
Beijing
Russia
Japan
Moscow
Tokyo
Turkey
Ankara
Poland
Germany
France
Warsaw
Berlin
Paris
Italy
Greece
Athens
Rome
Spain
Madrid
Portugal
Lisbon
2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2

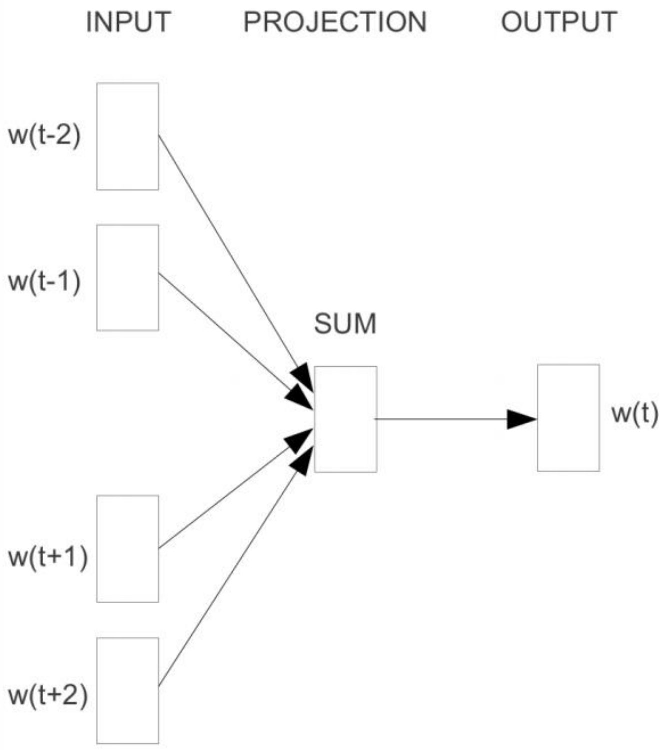
man is to woman as king is to ?



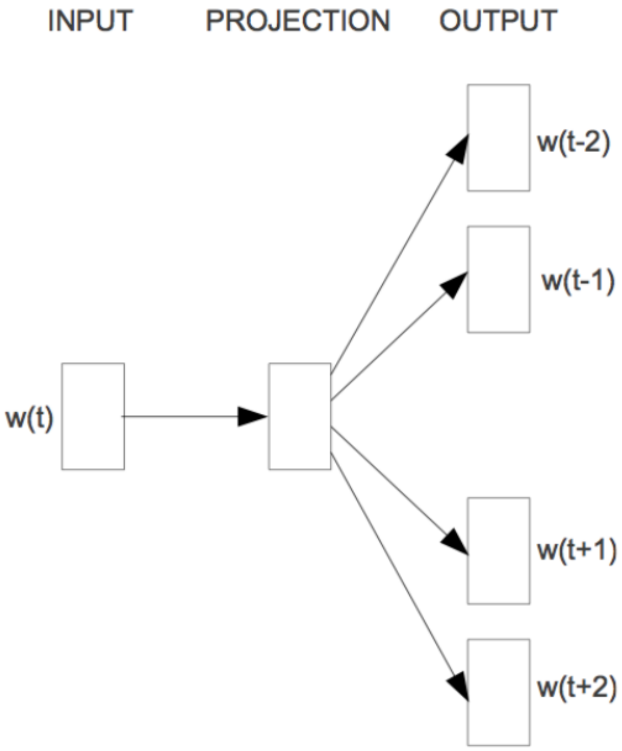
(Mikolov et al., NAACL HLT, 2013)

Word2vec: CBOW, skip-gram

- PageRank 알고리즘



CBOW



Skip-gram

- CBOW(Continuous Bag-Of-Words)
  - 좌우 문맥으로부터 현재 단어 예측
  - 학습속도 빠름, 대규모 학습말뭉치에 적합

- Continuous skip-gram
  - Skip-gram: n-gram with a gap(현재 단어)
  - 빈도가 높은 단어 벡터를 잘 표현함
  - 학습 속도 느림

Using word2vec

- Original: http://word2vec.googlecode.com/svn/trunk/
  - https://code.google.com/p/word2vec/
- C++11 version: https://github.com/jdeng/word2vec
- Python: http://radimrehurek.com/gensim/models/word2vec.html
- Java: https://github.com/ansjsun/word2vec_java
  - http://deeplearning4j.org/word2vec
- Parallel java: https://github.com/siegfang/word2vec
- CUDAversion: https://github.com/whatupbiatch/cuda-word2vec

https://code.google.com/archive/p/word2vec/

- word2vec - Revision 42: /trunk
  - LICENSE
  - README.txt
  - compute-accuracy.c
  - demo-analogy.sh
  - demo-classes.sh
  - demo-phrase-accuracy.sh
  - demo-phrases.sh
  - demo-train-big-model-v1.sh
  - demo-word-accuracy.sh
  - demo-word.sh
  - distance.c
  - makefile
  - questions-phrases.txt
  - questions-words.txt
  - word-analogy.c
  - word2phrase.c
  - word2vec.c

- Run 'make' to compile word2vec tool
- Run the demo scripts:
  - ./demo-word.sh
  - ./demo-analogy.sh
  - ./demo-phrases.sh

Ex) Word similarities in word2vec

Sweden
Similar words

Word	Cosine distance
norway	0.760124
denmark	0.715460
finland	0.620022
switzerland	0.588132
belgium	0.585835
netherlands	0.574631
iceland	0.562368
estonia	0.547621
slovenia	0.531408
floorball_federation	0.529570
luxembourg	0.529477
czech_republic	0.528778
slovakia	0.526340
romania	0.524281
kista	0.522488
helsinki_vantaa	0.519936
swedish	0.519901
balrog_ik	0.514556
portugal	0.502495
russia	0.500196
slovakia_slovenia	0.496051
ukraine	0.495712
chur	0.484225
thailand_togo	0.479172
crimea_ukraine	0.478596

Harvard
Similar words

Word	Cosine distance
yale	0.638970
cambridge	0.612665
university	0.597709
faculty	0.588422
harvey_mudd	0.578338
johns_hopkins	0.575645
graduate	0.570294
undergraduate	0.565881
professor	0.563657
mcgill	0.562168
ph_d	0.558665
california_berkeley	0.555539
yale_university	0.550480
harvard_crimson	0.549848
princeton	0.544070
college	0.542838
oxford	0.531948
barnard_college	0.530800
professors	0.529959
princeton_university	0.529763
ucl	0.527395
doctorates	0.526292
doctoral	0.523317
cambridge_massachusetts	0.519657
juris_doctor	0.518845
graduate_student	0.518815
postgraduate	0.515757

Interesting properties of word2vec

- vector('Paris') - vector('France') + vector('Italy’)
  is very close to vector('Rome')
- vector('king') - vector('man') + vector('woman’)
  is close to vector('queen')

- Simple demo by running demo-analogy.sh

How to measure

- Quality of the word vectors
  ./demo-word-accuracy.sh -- word relation test set
  ./demo-phrase-accuracy.sh -- phrase relation test set
- Word clustering
  ./demo-classes.sh

Where to obtain the training data

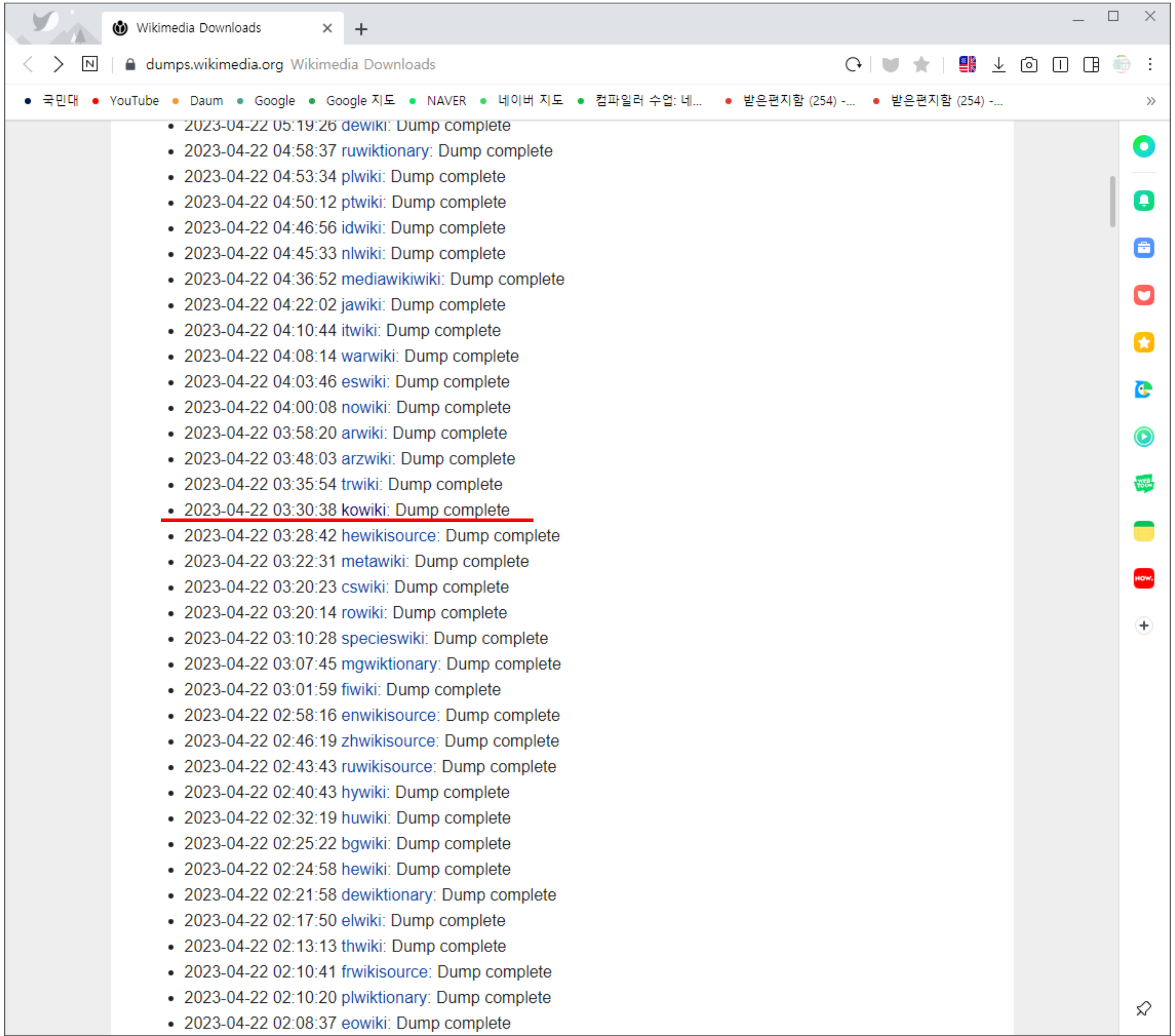
- Latest Wikipedia dump
  - https://dumps.wikimedia.org/backup-index.html

- WMT11 site: text data for several languages
  - http://www.statmt.org/wmt11/translation-task.html#download
  - Statistical M.T. -- http://statmt.org/

- UMBC webbase corpus around 3 billion words

- Polyglot
  - https://sites.google.com/site/rmyeid/projects/polyglot

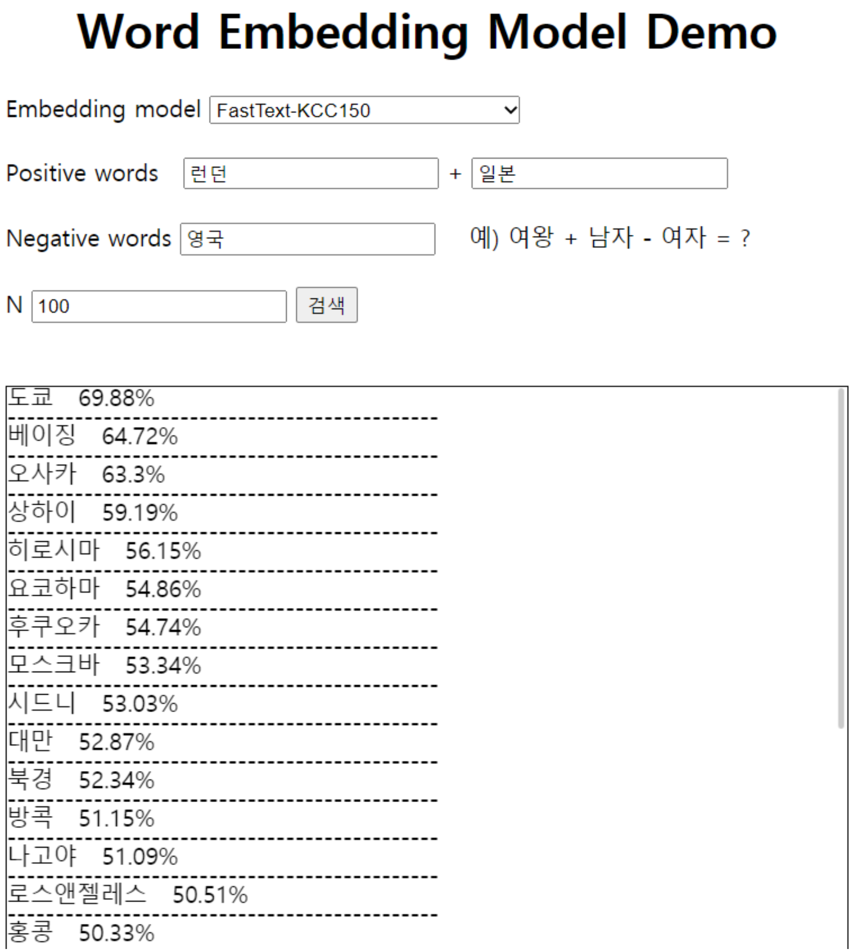
https://dumps.wikimedia.org/backup-index.html



Word Embedding 데모, 실습

- http://nlp.kookmin.ac.kr/kcc/word2vec/
- http://nlp.kookmin.ac.kr/kcc/word2vec/demo

- https://bitbucket.org/aboSamoor/word2embeddings
- https://sites.google.com/site/rmyeid/projects/polyglot
- https://remykarem.github.io/word2vec-demo/
- http://nbviewer.jupyter.org/gist/aboSamoor/6046170



Demo: similar words

https://sites.google.com/site/rmyeid/projects/polyglot

Online Demo

The demo shows words proximity in the embedding space. Given a word we calculate its neighbours in the space according to the Euclidean distance. In case, you are using the latest version of Firefox 23.0+, this demo will be blocked by default. Here are instructions on how to disable protection and enable the demo. Otherwise, you can have direct access to the demo at <wordrepresentation.appspot.com>.

Obama | English | Search

Examples:

- English
  - Bush vs bush
  - Apple vs apple
  - Mumbai vs Bombay
  - Apple vs Google
  - accelerating
- Russian
  - Москва (Moscow)
  - водка (Vodka)
  - Путин (Putin)
  - Пушкин (Pushkin)
  - USA

Rank	Word	L2 Distance
0	Apple	0.0
1	Dell	3.21376
2	Paramount	3.73771
3	Mac	3.75451
4	Flex	3.95375
5	Link	3.97127
6	Fox	4.12825
7	HP	4.13556
8	Oracle	4.24255
9	Cream	4.26531
10	Shell	4.29036
11	Dot	4.32141
12	AOL	4.3369
13	Safari	4.46203
14	Flash	4.55888
15	Bell	4.60953
16	Diamond	4.61724
17	Mercury	4.66379
18	AMC	4.686
19	Dial	4.78557
20	Rhino	4.79876

Rank	Word	L2 Distance
0	apple	0.0
1	tomato	2.1517
2	bean	2.28461
3	onion	2.32823
4	potato	2.33879
5	chicken	2.6566
6	chocolate	2.68221
7	lemon	2.70843
8	almond	2.73388
9	berry	2.77824
10	pepper	2.80792
11	strawberry	2.85157
12	chili	2.86959
13	sausage	2.88144
14	egg	2.90296
15	dessert	2.91029
16	peach	2.91455
17	mango	2.94624
18	turkey	3.00063
19	coconut	3.04677
20	pea	3.08457

Using it in Python

- Gensim Python Library
  https://radimrehurek.com/gensim/index.html

- Gensim Tutorials
  https://radimrehurek.com/gensim/tutorial.html

- Visualization -- Scikit-Learn t-SNE
  http://scikit-learn.org/stable/modules/generated/sklearn.manifold.TSNE.html

Word Embedding Methods

- Word2vec
- GloVe
- FastText
- ELMo
- BERT: ALBERT, RoBERTa
- XLNet
- ELECTRA
- GPT-2

- Evaluation(MRC): SQuAD, RACE, GLUE, etc

BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding

https://www.aclweb.org/anthology/N19-1423.pdf (NAACL'2019)

Code and pre-trained models: https://github.com/google-research/bert

Pre-trained Language Representations

- *Feature-based* approach : ELMo (Peters et al., 2018a)
    - pre-trained representations as additional features

- *Fine-tuning* approach : OpenAI GPT (Radford et al., 2018)
    - introduces minimal task-specific parameters
    - simply fine-tuning all pretrained parameters

- Two approaches use *unidirectional language models* to learn general language representations

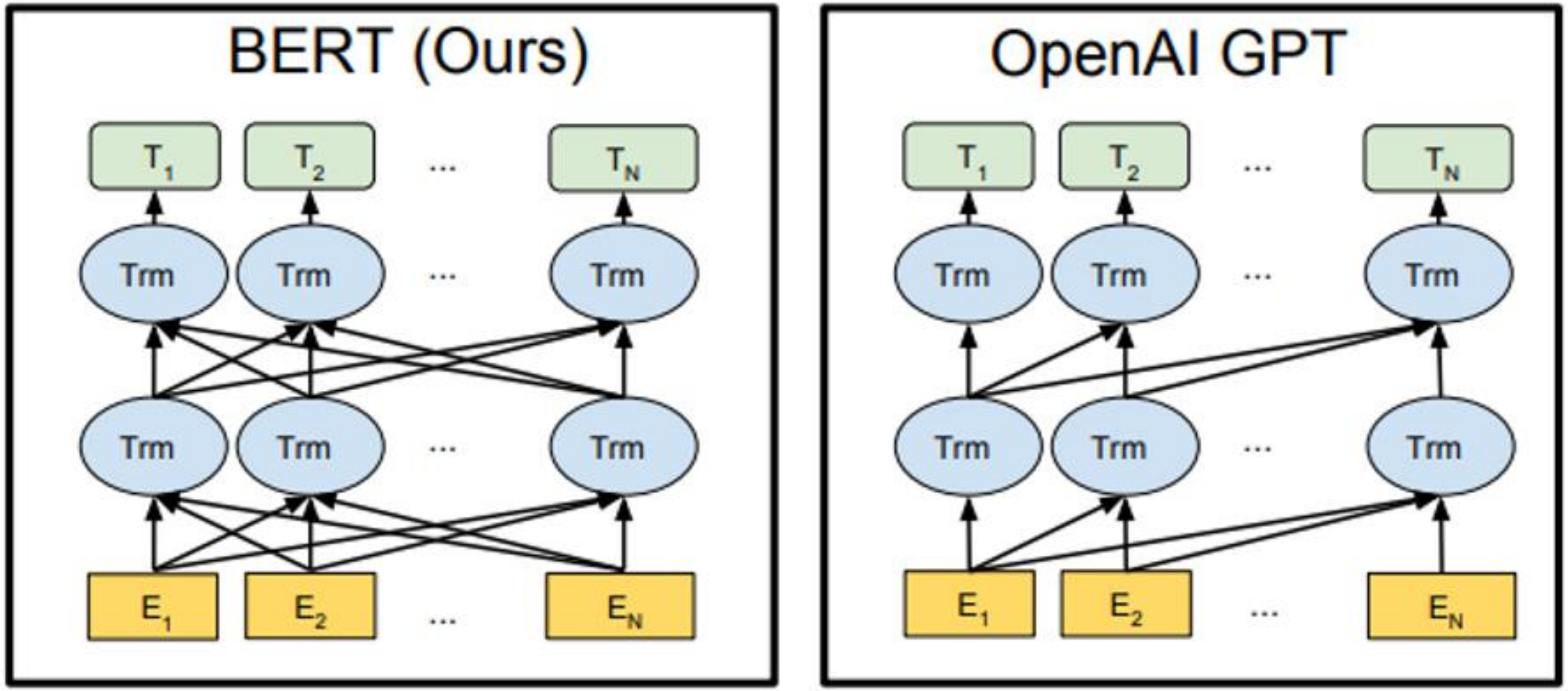
BERT: focused on *bidirectional pre-training*

- Improve the fine-tuning based approach by using

- "masked language model" (MLM)
  - randomly masks some of the tokens from the input,
  - and predicts the original vocabulary

- "next sentence prediction"
  - jointly pretrains text-pair representations.

BERT vs. GPT


BERT (Ours)
T1 T2 ... TN
Trm Trm ... Trm
Trm Trm ... Trm
E1 E2 ... EN
OpenAI GPT
T1 T2 ... TN
Trm Trm ... Trm
Trm Trm ... Trm
E1 E2 ... EN

State-of-the-art results on SQuAD and RACE

Models	SQuAD1.1 dev	SQuAD2.0 dev	SQuAD2.0 test	RACE test (Middle/High)
Single model (from leaderboard as of Sept. 23, 2019)				
BERT-large	90.9/84.1	81.8/79.0	89.1/86.3	72.0 (76.6/70.1)
XLNet	94.5/89.0	88.8/86.1	89.1/86.3	81.8 (85.5/80.2)
RoBERTa	94.6/88.9	89.4/86.5	89.8/86.8	83.2 (86.5/81.3)
UPM	-	-	89.9/87.2	-
XLNet + SG-Net Verifier++	-	-	90.1/87.2	-
ALBERT (1M)	94.8/89.2	89.9/87.2	-	86.0 (88.2/85.1)
ALBERT (1.5M)	**94.8/89.3**	**90.2/87.4**	**90.9/88.1**	**86.5 (89.0/85.5)**
Ensembles (from leaderboard as of Sept. 23, 2019)				
BERT-large	92.2/86.2	-	-	-
XLNet + SG-Net Verifier	-	-	90.7/88.2	-
UPM	-	-	90.7/88.2	
XLNet + DAAF + Verifier	-	-	90.9/88.6	-
DCMN+	-	-	-	84.1 (88.5/82.3)
ALBERT	**95.5/90.1**	**91.4/88.9**	**92.2/89.7**	**89.4 (91.2/88.6)**

Table 10: State-of-the-art results on the SQuAD and RACE benchmarks.

SQuAD   Home   Explore 2.0   Explore 1.1

SQuAD2.0

The Stanford Question Answering Dataset

What is SQuAD?

Stanford **Qu**estion **A**nswering **D**ataset (SQuAD) is a reading comprehension dataset, consisting of questions posed by crowdworkers on a set of Wikipedia articles, where the answer to every question is a segment of text, or *span*, from the corresponding reading passage, or the question might be unanswerable.

SQuAD2.0 combines the 100,000 questions in SQuAD1.1 with over 50,000 unanswerable questions written adversarially by crowdworkers to look similar to answerable ones. To do well on SQuAD2.0, systems must not only answer questions when possible, but also determine when no answer is supported by the paragraph and abstain from answering.

Explore SQuAD2.0 and model predictions

SQuAD2.0 paper (Rajpurkar & Jia et al. '18)

SQuAD 1.1, the previous version of the SQuAD dataset, contains 100,000+ question-answer pairs on 500+ articles.

Explore SQuAD1.1 and model predictions

SQuAD1.0 paper (Rajpurkar et al. '16)

Getting Started

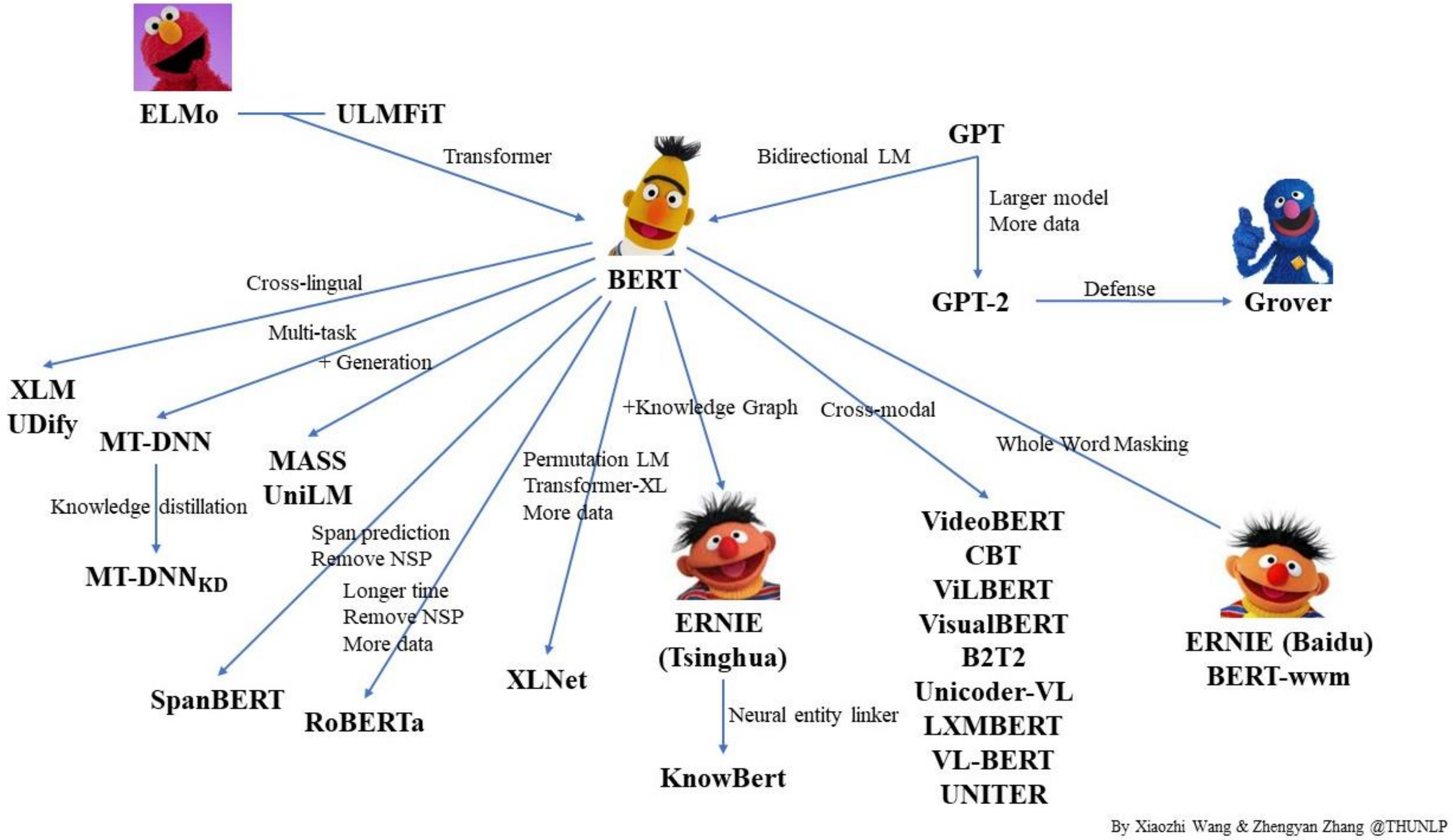
We've built a few resources to help you get started with the dataset.
Download a copy of the dataset (distributed under the CC BY-SA 4.0 license):

Leaderboard

SQuAD2.0 tests the ability of a system to not only answer reading comprehension questions, but also abstain when presented with a question that cannot be answered based on the provided paragraph.

Rank	Model	EM	F1
	Human Performance *Stanford University* (Rajpurkar & Jia et al. '18)	86.831	89.452
1 Jun 04, 2021	IE-Net (ensemble) *RICOH_SRCB_DML*	**90.939**	**93.214**
2 Feb 21, 2021	FPNet (ensemble) *Ant Service Intelligence Team*	90.871	93.183
3 May 16, 2021	IE-NetV2 (ensemble) *RICOH_SRCB_DML*	90.860	93.100
4 Apr 06, 2020	SA-Net on Albert (ensemble) *QIANXIN*	90.724	93.011
5 May 05, 2020	SA-Net-V2 (ensemble) *QIANXIN*	90.679	92.948
5 Apr 05, 2020	Retro-Reader (ensemble) *Shanghai Jiao Tong University* http://arxiv.org/abs/2001.09694	90.578	92.978
5 Feb 05, 2021	FPNet (ensemble) *YuYang*	90.600	92.899
6 Apr 18, 2021	TransNets + SFVerifier + SFEnsembler (ensemble) *Senseforth AI Research* https://www.senseforth.ai/	90.487	92.894
6 Dec 01, 2020	EntitySpanFocusV2 (ensemble) *RICOH_SRCB_DML*	90.521	92.824

ELMo, BERT, GPT



BERT의 성능 비교

	BERT	RoBERT	DistilBERT	XLNet
Size (millions)	**Base**: 110 **Large**: 340	**Base**: 110 **Large**: 340	**Base**: 66	**Base**: ~110 **Large**: ~340
Training Time	**Base**: 8 x V100 x 12 days* **Large**: 64 TPU Chips x 4 days (or 280 x V100 x 1 days*)	**Large**: 1024 x V100 x 1 day; 4-5 times more than BERT.	**Base**: 8 x V100 x 3.5 days; 4 times less than BERT.	**Large**: 512 TPU Chips x 2.5 days; 5 times more than BERT.
Performance	Outperforms state-of-the-art in Oct 2018	2-20% improvement over BERT	5% degradation from BERT	2-15% improvement over BERT
Data	16 GB BERT data (Books Corpus + Wikipedia). 3.3 Billion words.	160 GB (16 GB BERT data + 144 GB additional)	16 GB BERT data. 3.3 Billion words.	**Base**: 16 GB BERT data **Large**: 113 GB (16 GB BERT data + 97 GB additional). 33 Billion words.
Method	BERT (Bidirectional Transformer with MLM and NSP)	BERT without NSP**	BERT Distillation	Bidirectional Transformer with Permutation based modeling

Comparison of BERT and recent improvements over it

BERT vs XLNet

Model	MNLI	QNLI	QQP	RTE	SST-2	MRPC	CoLA	STS-B	WNLI
Single-task single models on dev									
BERT [2]	86.6/-	92.3	91.3	70.4	93.2	88.0	60.6	90.0	-
XLNet	**89.8/-**	**93.9**	**91.8**	**83.8**	**95.6**	**89.2**	**63.6**	**91.8**	-
Single-task single models on test									
BERT [10]	86.7/85.9	91.1	89.3	70.1	94.9	89.3	60.5	87.6	65.1
Multi-task ensembles on test (from leaderboard as of June 19, 2019)									
Snorkel* [29]	87.6/87.2	93.9	89.9	80.9	96.2	91.5	63.8	90.1	65.1
ALICE*	88.2/87.9	95.7	**90.7**	83.5	95.2	92.6	**68.6**	91.1	80.8
MT-DNN* [18]	87.9/87.4	96.0	89.9	**86.3**	96.5	92.7	68.4	91.1	89.0
XLNet*	**90.2/89.7**†	**98.6**†	90.3†	**86.3**	**96.8**†	**93.0**	67.8	**91.6**	**90.4**

Source: https://arxiv.org/abs/1906.08237

References

- Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient Estimation of Word Representations in Vector Space. In Proceedings of Workshop at ICLR, 2013.
- Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. Distributed Representations of Words and Phrases and their Compositionality. In Proceedings of NIPS, 2013.
- Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. Linguistic Regularities in Continuous Space Word Representations. In Proceedings of NAACL HLT, 2013.
- GloVe: Global Vectors for Word Representation
- XLNet: Generalized Autoregressive Pretraining for Language Understanding
- SQuAD -- https://web.stanford.edu/class/cs224n/slides/cs224n-2019-lecture10-QA.pdf
- GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding

정리

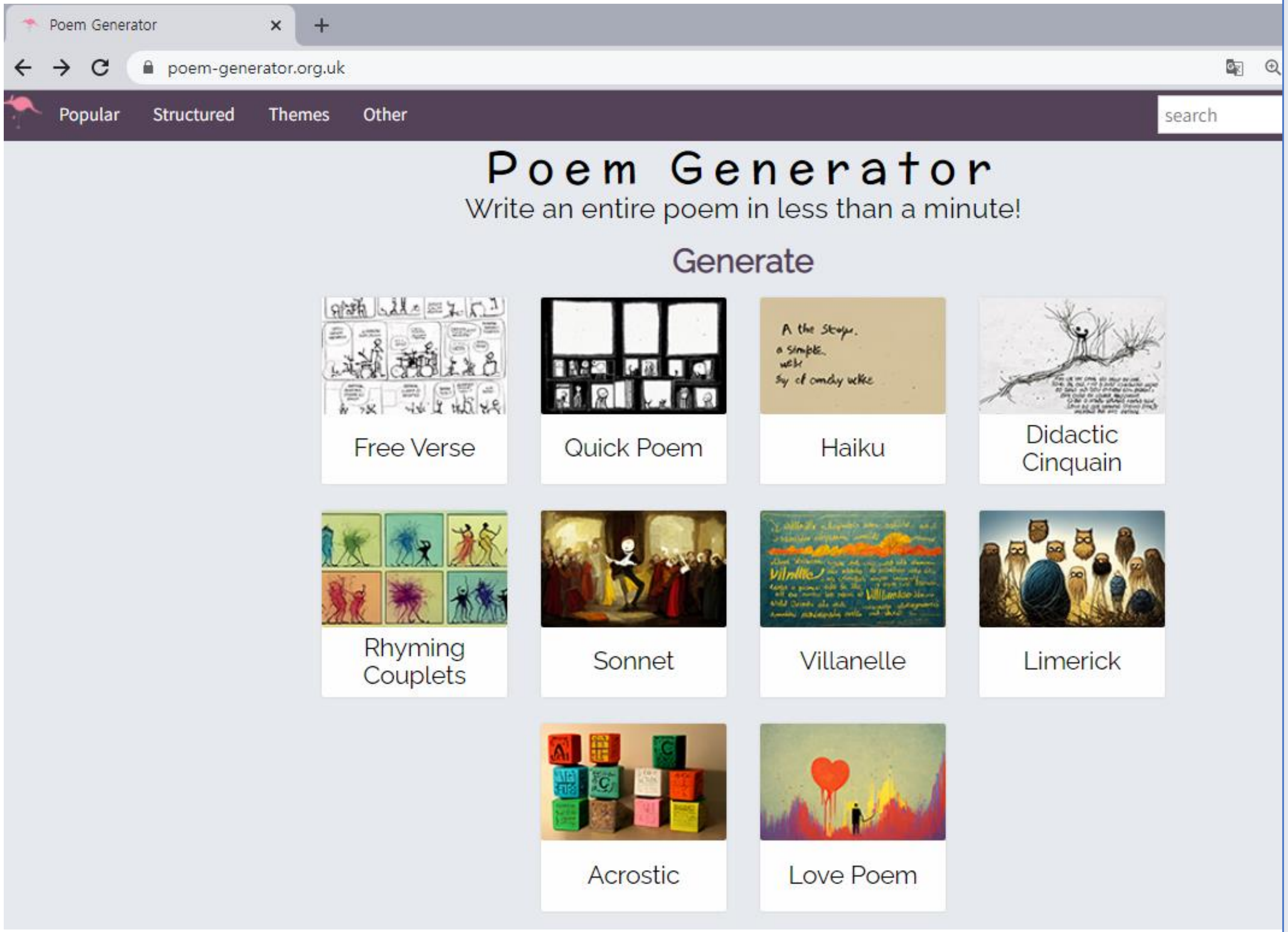
1. 워드 임베딩은 어떻게 활용되는가?

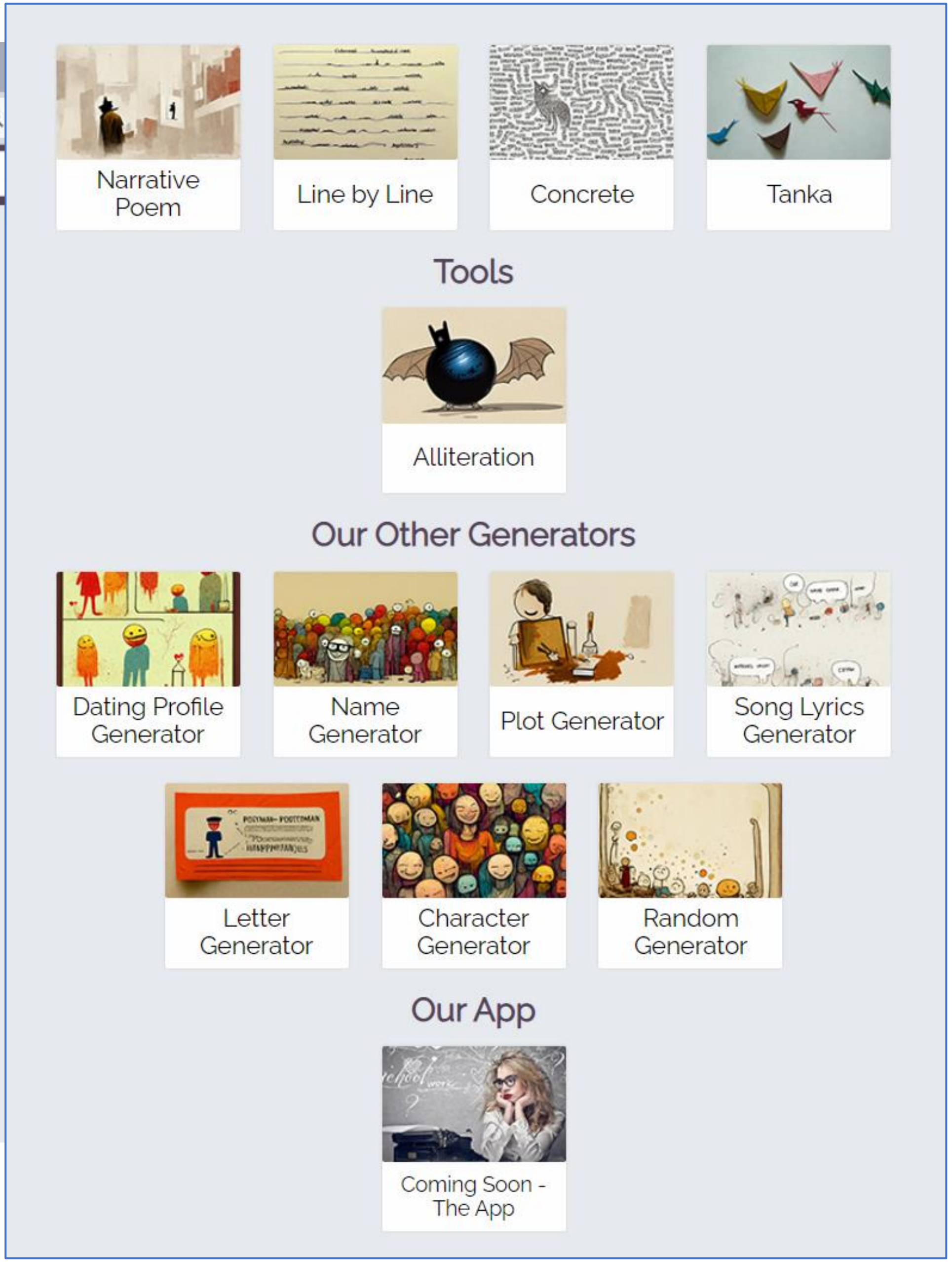
2. 형태소 분석(또는 품사 태깅)의 필요성은 무엇인가?

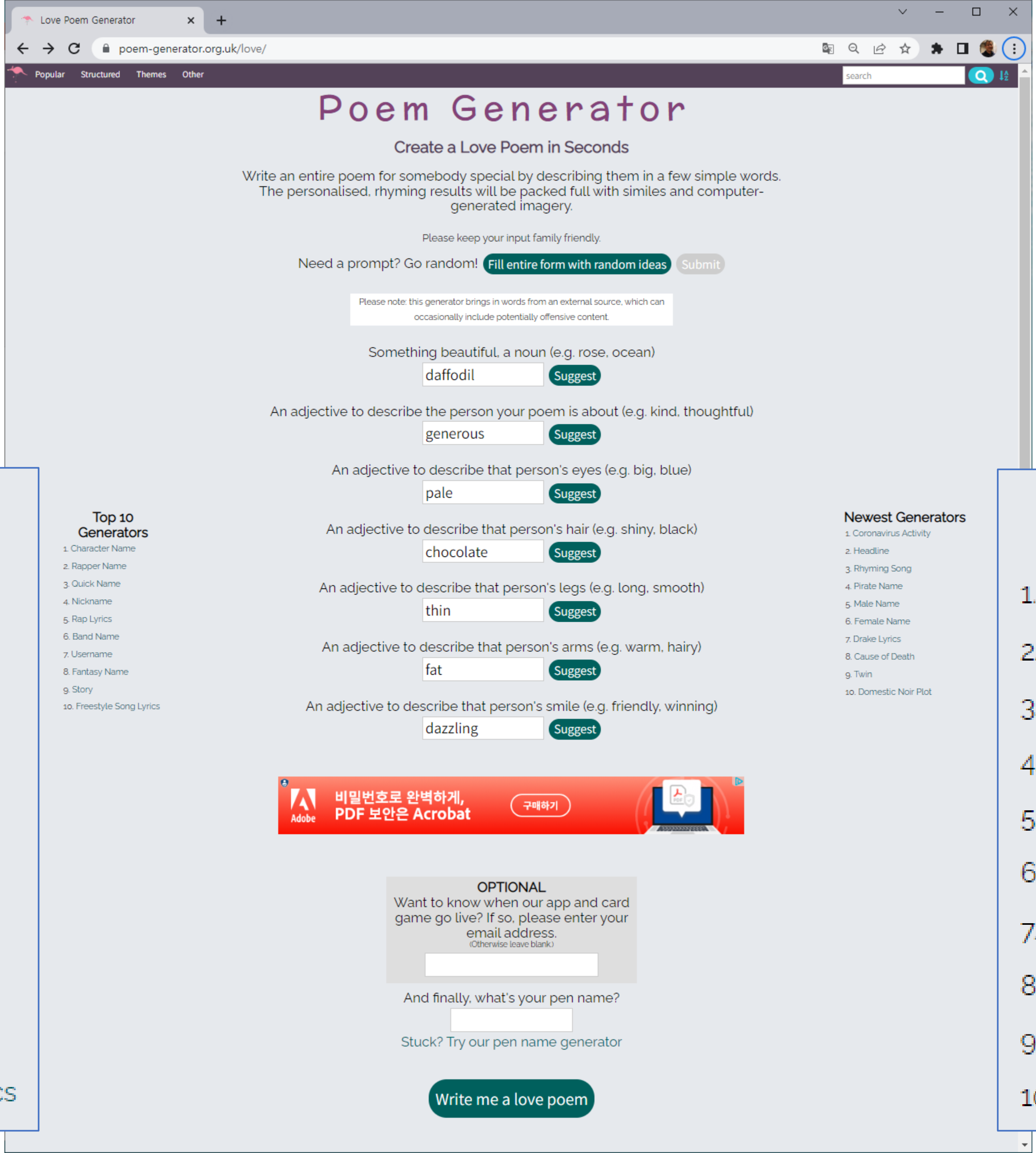
3. Subword 토크나이저의 필요성은 무엇인가?

4. BERT 모델의 특징 및 활용 분야는 무엇인가?

https://www.poem-generator.org.uk/







Top 10 Generators

1. Character Name
2. Rapper Name
3. Quick Name
4. Nickname
5. Rap Lyrics
6. Band Name
7. Username
8. Fantasy Name
9. Story
10. Freestyle Song Lyrics

Newest Generators

1. Coronavirus Activity
2. Headline
3. Rhyming Song
4. Pirate Name
5. Male Name
6. Female Name
7. Drake Lyrics
8. Cause of Death
9. Twin
10. Domestic Noir Plot