CIS530 Spring 2018 Assignment 4
Yezheng Li (02/07)

---

1 Sparse Vector Representations

We tried X variations of a sparse vector representation. The variations we tried were:

- cooccurrence term-context matrix I made myself with D = 500, window =3 (0.3730)
- cooccurrence term-context matrix I made myself with D = 500, window =4 (0.3711)
- cooccurrence term-context matrix I made myself with D = 500, window =5 (0.3701)
- cooccurrence term-context matrix I made myself with D = 1000, window =3 (0.3702)
- concatenate of the two (with appropriate trunction if necessary). Truncation is also attempted which tries to avoid curse of dimensionality – in order to distinguish this case from previous variations, truncation is made the same for GLOVES and Word2Vec – for example, just take the first 200 features of 800-3 and the first 400 features of 500-4.

We used KMeans, SpectralClustering, GMM as the clustering algorithm:

In table 1,

- KMeans means I just use KMeans;
- SpectralClustering in sparse case typically result in bug – my guess is the graph is not far from connected (even with keneralized geometry);
- KMeans + GMM means I use GMM but corrected (with randomness) by KMeans: correction is necessary because GMM (as well as SpectralClustering) may frequently result in empty clusters: take 'expected.v' as an example, although I assign number of clusters 6, GMM (as well as SpectralClustering) may result in empty clusters for certain labels.

Vector Space Model	KMeans	KMeans+GGM
Baseline (provided)	0.3653	0.3636 – 0.3701
cooccurrence matrix 500-3(I made myself)	0.3710	0.3579 - 0.3661
cooccurrence matrix 500-4(I made myself)	0.3723	0.3691 - 0.3722
cooccurrence matrix 500-5(I made myself)	0.3705	0.3651 - 0.3717
cooccurrence matrix 600-3	0.3701	0.3663 - 0.3707
cooccurrence matrix 1000-3	0.3706	0.3724 – 0.3737
1000-3 + 500-4	0.3708	0.3689 – 0.3723

Table 1: Paired F-Score on the dev set by different vector space models and clustering algorithms.

We compare the impact of our model and clustering choices by evaluating their performance over the dev set (Table 1):

2 Dense Vector Representations

We tried > 10 variations of the dense vector representation. The variations we tried were:

- GLOVES50-300 (for GLOVES100);
- with truncated dimensions of features – it is a general case that fewer features lead to worse performance.
- GLOVES + Word2Vec. Truncation is also attempted which tries to avoid curse of dimensionality – in order to distinguish this case from previous variations, truncation is made the same for GLOVES and Word2Vec – for example, just take the first 200 features of GLOVES and the first 200 features of Word2Vec.

We used KMeans, SpectralClustering, GMM as the clustering algorithm.

In table 2,

- KMeans means I just use KMeans;

Dense Model	KMeans	KMeans+SpectralCluster	KMeans + GMM
Word2Vec (provided)	0.3251	0.3360 – 0.3461	0.3307 – 0.3462
GLOVES100	0.2989	(bugs not fixed)	0.2931 – 0.3117
GLOVES200	0.3280	0.32970 - 0.3377	0.3261 - 0.3320
GLOVES300	0.3280	0.3413 – 0.3435	0.3393 – 0.3554
GLOVES300 +Word2Vec	0.3138	0.3367-0.3429	0.3327 – 0.3473

Table 2: Paired F-Score on the dev set by different dense vector space models and clustering algorithms.

- KMeans + SpectralClustering means I use SpectralClustering but corrected (with randomness) by KMeans: correction is necessary because SpectralClustering may frequently result in empty clusters: take 'expected.v' as an example, although I assign number of clusters 6, GMM (as well as SpectralClustering) may result in empty clusters for certain labels.
- KMeans + GMM means I use GMM but corrected (with randomness) by KMeans: correction is necessary because GMM may frequently result in empty clusters: take 'expected.v' as an example, although I assign number of clusters 6, GMM may result in empty clusters for certain labels.

We compare the impact of our model and clustering choices by evaluating their performance over the dev set (Table 2).

3 Comparison

Overall, our cooccurrence matrix 1000-3 (with KMeansGMM) model, a sparse model, performs best.

In general, we can compare dense and sparse models by looking at instances where one does well and the other fails.

Examples of target words where the dense model scores high and the sparse model scores low are... (describe what they have in common, hypothesize why this might be the case) high paper.n (0.6747) with oracle:

paper.n :: 1 :: newspaper product production
paper.n :: 2 :: newspaper publisher
paper.n :: 3 :: rag tabloid sheet daily newspaper gazette press
paper.n :: 4 :: essay report theme composition
paper.n :: 5 :: manifold chad manilla tissue flypaper parchment sheet tablet newsprint pad oilpaper manila papier-mache wallpaper paper-mache blotter material confetti carbon cardboard card crepe papyrus linen stuff newspaper
paper.n :: 6 :: medium
paper.n :: 7 :: article

while I cluster it into

paper.n :: 1 :: tissue article paper-mache crepe papyrus theme manila chad sheet manilla newsprint tabloid linen carbon flypaper publisher pad material essay confetti parchment papier-mache cardboard card gazette tablet oilpaper medium stuff blotter manifold wallpaper composition rag
paper.n :: 2 :: report
paper.n :: 3 :: production
paper.n :: 4 :: daily
paper.n :: 5 :: product
paper.n :: 6 :: press
paper.n :: 7 :: newspaper

note.v (0.6400) with oracle

note.v :: 1 :: observe
note.v :: 2 :: comment remark state notice mention say observe tell
note.v :: 3 :: notice mark

while I cluster into note.v :: 1 :: say
note.v :: 2 :: comment mark tell observe notice remark mention
note.v :: 3 :: state

low play.v (0.1940) with oracle

play.v :: 1 :: confront face replay encounter meet
play.v :: 2 :: hook hit

play.v :: 3 :: use utilise utilize employ apply promote
play.v :: 4 :: employ apply use utilize utilise
play.v :: 5 :: tucker exhaust beat
play.v :: 6 :: assume feign simulate sham
play.v :: 7 :: sound
play.v :: 8 :: underplay gamble
play.v :: 9 :: recreate
play.v :: 10 :: do toy behave act
play.v :: 11 :: move act
play.v :: 12 :: perform
play.v :: 13 :: deploy cover declare pitch
play.v :: 14 :: run
play.v :: 15 :: do behave act
play.v :: 16 :: portray represent make pretend emote re-create impersonate act reenact support enact parody
play.v :: 17 :: flirt toy act dally move
play.v :: 18 :: trifle dally consider deal take
play.v :: 19 :: retire diddle toy fiddle manipulate
play.v :: 20 :: raise stake game back gage see punt gamble wager bet
play.v :: 21 :: move displace
play.v :: 22 :: bring create work make wreak
play.v :: 23 :: golf vie foul walk nail seesaw unblock revoke curl contend putt develop field bowl fullback start cradle exit fumble croquet call quarterback cricket bandy complete bid teeter-totter teetertotter snooker catch backstop ace die replay compete gamble volley misplay
play.v :: 24 :: debut
play.v :: 25 :: accompany perform bow busk swing follow symphonize symphonise
play.v :: 26 :: move act
play.v :: 27 :: fool disport rollick cavort romp frolic dabble paddle skylark frisk act lark roughhouse sport gambol
play.v :: 28 :: sound chord drum beat register clarion pipe tweedle fiddle skirl trumpet harp play.v :: 29 :: rag bugle repeat reprize slur misplay jazz reprise recapitulate spiel re-create replay fiddle tongue riff prelude modulate
play.v :: 30 :: go travel move locomote
play.v :: 31 :: underplay overplay ham perform stooge underact mime overact act play-act roleplay pantomime
play.v :: 32 :: exploit work
play.v :: 33 :: wager bet

play.v :: 34 :: discharge

while I cluster it into

play.v :: 1 :: foul putt modulate fullback pitch bandy recapitulate dabble encounter frolic symphonise promote accompany dally sham croquet slur discharge perform romp teetertotter retire prelude emote skylark punt die harp toy manipulate ham catch underact flirt displace vie paddle compete cricket bet tucker quarterback misplay underplay mime reprise disport pantomime fool roughhouse cavort portray replay backstop drum tweedle lark reprize parody unblock rag roleplay swing fiddle confront re-create walk overact spiel exploit pretend pipe exhaust declare rollick feign gambol symphonize curl utilise reenact stooge utilize contend wreak clarion overplay trifle bugle cradle gamble behave assume playact frisk impersonate revoke hook golf repeat teeter-totter busk tongue fumble ace enact gage recreate sport wager chord snooker exit locomote nail volley seesaw simulate diddle bow sound deploy employ riff jazz trumpet bowl skirl
play.v :: 2 :: see
play.v :: 3 :: face
play.v :: 4 :: make
play.v :: 5 :: move
play.v :: 6 :: support
play.v :: 7 :: back
play.v :: 8 :: deal
play.v :: 9 :: stake
play.v :: 10 :: raise
play.v :: 11 :: start
play.v :: 12 :: bid
play.v :: 13 :: do
play.v :: 14 :: take
play.v :: 15 :: work
play.v :: 16 :: meet
play.v :: 17 :: game
play.v :: 18 :: call
play.v :: 19 :: hit
play.v :: 20 :: beat
play.v :: 21 :: go
play.v :: 22 :: use
play.v :: 23 :: run
play.v :: 24 :: travel represent consider apply

play.v :: 25 :: cover
play.v :: 26 :: create
play.v :: 27 :: bring
play.v :: 28 :: field
play.v :: 29 :: develop
play.v :: 30 :: register
play.v :: 31 :: follow
play.v :: 32 :: act
play.v :: 33 :: complete
play.v :: 34 :: debut

bank.n (0.2143) with the oracle

bank.n :: 1 :: depository deposit repository depositary
bank.n :: 2 :: acquirer
bank.n :: 3 :: stockpile backlog reserve
bank.n :: 4 :: array
bank.n :: 5 :: slope cant camber incline side
bank.n :: 6 :: funds finances
bank.n :: 7 :: slope incline riverbank riverside side waterside
bank.n :: 8 :: ridge sandbank bluff
bank.n :: 9 :: container

while I cluster it into

bank.n :: 1 :: deposit
bank.n :: 2 :: slope
bank.n :: 3 :: funds
bank.n :: 4 :: side
bank.n :: 5 :: reserve
bank.n :: 6 :: finances
bank.n :: 7 :: riverside acquirer stockpile depositary depository waterside riverbank repository bluff array ridge camber cant sandbank incline
bank.n :: 8 :: container
bank.n :: 9 :: backlog

- **Hypothesis:** the larger number of clusters, the more difficult the task – take a look at "play.v", "paper.n", "note.v" for evidences. In addition, I think my algorithm has tendency of unevenly distributed result, which explains my clustering result for "play.v", "bank.n".

Examples of target words where the sparse model scores high and the dense model scores low are... (describe what they have in common, hypothesize why this might be the case)

High note.v (0.6400) has oracle

note.v :: 1 :: observe
note.v :: 2 :: comment remark state notice mention say observe tell
note.v :: 3 :: notice mark

with my result

note.v :: 1 :: mark tell observe state notice say mention
note.v :: 2 :: comment
note.v :: 3 :: remark

plan.n (0.6279) has the oracle

plan.n :: 1 :: elevation drawing
plan.n :: 2 :: blueprint schema regimen scheme projection schedule pattern outline budget idea project program design thought agenda guideline docket programme regime
plan.n :: 3 :: layout organisation system arrangement design trap organization snare configuration constellation

with my result

plan.n :: 1 :: snare trap
plan.n :: 2 :: organization outline project pattern blueprint scheme idea thought system organisation regimen docket schedule projection drawing arrangement guideline regime budget constellation agenda programme program
plan.n :: 3 :: layout design schema elevation configuration

Low play.v (0.1433) with oracle

play.v :: 1 :: confront face replay encounter meet
play.v :: 2 :: hook hit

play.v :: 3 :: use utilise utilize employ apply promote
play.v :: 4 :: employ apply use utilize utilise
play.v :: 5 :: tucker exhaust beat
play.v :: 6 :: assume feign simulate sham
play.v :: 7 :: sound
play.v :: 8 :: underplay gamble
play.v :: 9 :: recreate
play.v :: 10 :: do toy behave act
play.v :: 11 :: move act
play.v :: 12 :: perform
play.v :: 13 :: deploy cover declare pitch
play.v :: 14 :: run
play.v :: 15 :: do behave act
play.v :: 16 :: portray represent make pretend emote re-create impersonate act reenact support enact parody
play.v :: 17 :: flirt toy act dally move
play.v :: 18 :: trifle dally consider deal take
play.v :: 19 :: retire diddle toy fiddle manipulate
play.v :: 20 :: raise stake game back gage see punt gamble wager bet
play.v :: 21 :: move displace
play.v :: 22 :: bring create work make wreak
play.v :: 23 :: golf vie foul walk nail seesaw unblock revoke curl contend putt develop field bowl fullback start cradle exit fumble croquet call quarterback cricket bandy complete bid teeter-totter teetertotter snooker catch backstop ace die replay compete gamble volley misplay
play.v :: 24 :: debut
play.v :: 25 :: accompany perform bow busk swing follow symphonize symphonise
play.v :: 26 :: move act
play.v :: 27 :: fool disport rollick cavort romp frolic dabble paddle skylark frisk act lark roughhouse sport gambol
play.v :: 28 :: sound chord drum beat register clarion pipe tweedle fiddle skirl trumpet harp play.v :: 29 :: rag bugle repeat reprize slur misplay jazz reprise recapitulate spiel re-create replay fiddle tongue riff prelude modulate
play.v :: 30 :: go travel move locomote
play.v :: 31 :: underplay overplay ham perform stooge underact mime overact act play-act roleplay pantomime
play.v :: 32 :: exploit work
play.v :: 33 :: wager bet

play.v :: 34 :: discharge

while I cluster it into

play.v :: 1 :: bugle
play.v :: 2 :: modulate recapitulate promote develop manipulate displace portray unblock confront use exploit reenact utilize wreak create enact recreate simulate deploy employ
play.v :: 3 :: punt fumble
play.v :: 4 :: vie
play.v :: 5 :: bandy dabble frolic dally slur romp emote skylark flirt paddle tucker mime reprise disport pantomime fool roughhouse cavort tweedle lark parody rag roleplay overact spiel pretend rollick gambol curl stooge trifle tongue diddle riff skirl
play.v :: 6 :: pitch game
play.v :: 7 :: underplay overplay
play.v :: 8 :: work do accompany perform retire travel die compete raise see go represent follow bring walk declare behave assume consider act start make take move meet apply
play.v :: 9 :: misplay
play.v :: 10 :: discharge
play.v :: 11 :: encounter hit symphonise run teetertotter prelude toy ham catch underact deal bid cover face back backstop reprize debut swing re-create complete support pipe exhaust symphonize utilise contend playact hook repeat teeter-totter ace call gage beat sport exit locomote nail register bow sound field
play.v :: 12 :: impersonate
play.v :: 13 :: golf
play.v :: 14 :: cradle
play.v :: 15 :: foul
play.v :: 16 :: chord
play.v :: 17 :: frisk
play.v :: 18 :: clarion
play.v :: 19 :: harp drum fiddle trumpet
play.v :: 20 :: bowl
play.v :: 21 :: busk
play.v :: 22 :: feign
play.v :: 23 :: putt
play.v :: 24 :: volley
play.v :: 25 :: seesaw

play.v :: 26 :: fullback quarterback
play.v :: 27 :: revoke
play.v :: 28 :: cricket snooker
play.v :: 29 :: jazz
play.v :: 30 :: croquet
play.v :: 31 :: replay
play.v :: 32 :: bet gamble wager
play.v :: 33 :: sham
play.v :: 34 :: stake

- **Hypothesis:** the larger number of clusters, the more difficult the task – take a look at "play.v", "plan.n", "note.v" for evidences.

  Dense model seems make the cluster more even (than sparse model; I like the phenomenon) although worse F-score . My hypothesis of such contradiction is because most of the words have contentrated meaning (I mean the cluster should be uneven somehow).

4 (Optional) Choosing K

We designed a model that chose the best K as follows...

Our results on the `test_nok_input.txt` data are given in (table below)...