3.2 Word Embedding

Training Corpora

We use 109 popular boards in PTT [5] and Chinese Wiki as our corpora.

Corpora	Words	Vocabulary size (frequency > 29)	Capacity
PTT,Wiki	858,118,783	249571	5.38 GB

Table 3.1: Word embedding training corpora.

Corpus Preprocessing

Remove tags, punctuations, numbers, non-Chinese, invalid and duplicate characters. There are some fixed format in different boards, such as opening hours of stores, the number of products or the reports written by which reporter. These words should be removed because they are not related to content. Because the fixed word length of one line (about 40 words), a sentence may be separated to multiple lines. Recombine these separated words back to original words if they are in vocabulary table.

Stop words removal Use stop words to remove some of content words and function words which have little semantic meaning. Content words include numerals and quantifiers. Numerals and quantifiers don't affect the main se-

[5] Taiwanese Bulletin Board System(BBS) which consists of 13,309 boards in plain text.

mantic meaning of the concepts even though they do have meanings. Although the number of objects has slightly different in context, they are still the same thing. It doesn't change any of its physical properties. For example, someone can drive a car, but can't drive two cars simultaneously. The main semantic meaning of a car or two cars are both cars. The number of cars doesn't change any property of it. It still has four tires, headlights, car doors, engines, and so on.

Some adverbs like adverbs of degree or time can be filtered out. Adverbs of degree are used to strengthen the meaning of adverbs or adjectives. Adverbs of time are used to describe when or how often an action happened. A concept with or without these words are basically the same. For example, this dress looks elegant and this dress looks very elegant. I buy a cake today and I bought a cake yesterday. It's not important how intensity an action, an adverb or an adjective. It's also not important when did you buy something. The point is how concepts interact to each other. Remove these types of adverbs may lose some information within sentence or paragraph, but the relations between concepts are still retained.

Some adverbs like adverbs of frequency can't be filtered out. For example, kids are always late for school and kids are seldom late for school. It's totally different how often kids late for school. It will change concept semantic meaning if they are removed.

Function words include conjunctions, particles, prepositions, interjection and onomatopoeia These function words have little concrete semantic meaning and for grammar purposes such as 和、的、從、啊、砰. They are used to make sentence grammatically correct and relate to other words in grammatical relation. They don't change the relations between concepts. For example, the students study in a classroom. It doesn't matter which specific classroom they are studying in. The relation between students and classroom is still "AtLocation" when function word "a" is filtered out.

Function words may decrease the performance of word embeddings. If there are many function words in a sentence, it would become fragment after segmenting without removing them. The strength of relation between content words without removing function words may lower than the sentence segmented after removing. It's possible that most of the content words relate to function words because of their high frequency in paragraph. It's intuitive to know that collocated words have stronger relation than others. Function words may relate to each content word, and this make them meaningless in word embeddings. It's important to select stop words. Have to consider each situation whether stop words have semantic meaning or not.

Text segmentation We use Jieba[6] API to segment words. Although it can find unknown words, which are not in self-defined dictionary, by a HMM-based (Hidden Markov Model) Viterbi algorithm, it also finds lots of unreasonable words, e.g., 裡你寫, 講你領, 將你傷. The distinct words contain "你" are found 9598 times in the result of discovering new words, and the other one are found 126 times. Therefore, we don't use HMM to combine unknown words. Exclude word frequency < 30 to reduce the impact of unreasonable

[6] https://github.com/fxsjy/jieba

words and rare words. We recombine segmented words back to their original word, e.g., [喝 了 酒] → [喝] [酒] → [喝 酒]. "了" is a stop words [喝] and [酒] are separated because of removing a stop word "了". Recombine these separated words so that it can represent its original meaning.

Count-based

Co-occurrence Matrix Build word-context co-occurrence matrix which is sparse and symmetric. In order to save the memory, we store the co-occurrence frequency only in upper triangular matrix. It saves half of the memory and speed up the computation.

The importance between target word and each context word is not the same. The closer to target word the more important it is. We adopt linear distance weighting in (3.1). Weight decreases as distance increases from target word.

$$[\ldots, target, 1, \frac{d-1}{d}, \frac{d-2}{d}, \ldots, \frac{1}{d}] \quad \text{d:distance to target word} \tag{3.1}$$

Weighted Co-occurrence Matrix Instead of using raw co-occurrence frequency, we use pointwise mutual information (PMI) [80] in (3.2) to weight.

$$PMI(x,y) = \log \frac{p(x,y)}{p(x)p(y)} = \log \frac{NC(x,y)}{C(x),C(y)} \tag{3.2}$$

$p(x)$: occurrence probability of word x
$p(x,y)$: co-occurrence probability of word x and y
$C(x)$: occurrence of x, and N is total number of words in corpus
$C(x,y)$: co-occurrence of x and y
N : total number of words in corpus

It calculates how much more x and y co-occur. The range of PMI is from $-\infty$ to $min[-\log p(x), -\log p(y)]$. The value is 0 if x and y is independent (co-occur by chance) in corpora. The value is maximized when x and y are perfectly associated (always co-occur), and the value is minimized when x and y barely co-occur. Words with strong association have higher PMI value (high C(x,y)). PMI has different variants, such as positive PMI (PPMI) [81], $\mathrm{PMI}^{\mathrm{k}}$ [82], Normalized PMI (NPMI) [83] and shifted PPMI(SPPMI) [84]. PPMI sets negative PMI value to 0. It makes sense to mark low correlation(tend not to co-occur) and uncorrelated (never co-occur) word pair to 0. The weighted matrix becomes more sparse since negative values are removed, and the computation cost is lower than PMI. $\mathrm{PMI}^{\mathrm{k}}$ and NPMI were proposed to make PMI less sensitive to rare words. Levy and Goldberg [84] found that skip-gram with negative sampling (SGNS) is implicitly factorizing a shifted PMI co-occurrence matrix, hence SPPMI is the original PMI shifted by $\log k$ (k $>$ 0).

$$SPPMI(x, y) = Max(PMI(x, y) - \log k, 0) \tag{3.3}$$

Dimensionality Reduction The size of our weighted matrix is 249571×249571 which is extremely large. We use truncated Singular Value Decomposition (truncated SVD) to reduce high-dimensional matrix. The formula of SVD and truncated SVD is in (3.4). $X_{m \times n}$ is a $m \times n$ co-occurrence matrix. $U_{m \times m}$ and $V^T_{n \times n}$ is left and right singular vectors. $\Sigma_{m \times n}$ is a diagonal matrix containing non-negative real number (singular values) on the diagonal in descending order. Truncated SVD discards values except first r largest singular values, they contain most of information in original matrix. It ensures the minimal loss of information and dimensionality reduction as well.

$$X_{m\times n} = U_{m\times m}\Sigma_{m\times n}V^T_{n\times n} \rightarrow U_{m\times r}\Sigma_{r\times r}V^T_{r\times n} \tag{3.4}$$

Weighting Exponent Caron p-transform [85] adjusts the scale of singular values in diagonal matrix Σ by weighting exponent p, where $X = U\Sigma^p$

Principal Components Removal Another method similar to Caron p-transform is principal components removal (PC-removal) [86]. It removes the first k dimensions of Σ. High variance dimensions may contain more unuseful information to lexical semantic tasks in contrast to low variance dimensions. If mapping the first k dimensions back to the co-occurrence space, most contributing words are people's names, "and" and "or". Figure 3.4[7] shows the the performance of dimensions removal in different levels of random noise. The performance falls off slowly (some information lost) if removing dimensions from matrix with a small amount of noise. It shows significant effect of noise reduction if removing dimensions from matrix with large amount of noise. Optimal value of weighting exponent p and the number of removed first k dimensions depending on tasks and corpus.

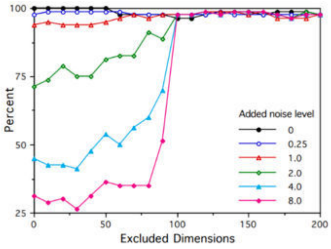

Figure 3.4: Dimensions removal.

[7] This figure is from [86]'s experiments.

Matrix Normalization Normalize the truncated SVD matrix to speed up the computation. Each row vector is unit vector (length = 1), therefore, the cosine similarity is dot product of two word embeddings.

After weighted co-occurrence matrix and dimensionality reduction, the count-based word embedding can be formulated in (3.5)

$$EM = [SVD_p(WCM)]_{m \times k:r} \tag{3.5}$$

EM	: embedding matrix
WCM	: weighted co-occurrence matrix
p	: weighting exponent
m	: number of rows (words)
k	: remove first k dimensions
r	: number of columns after truncated SVD

As to prediction-based word embedding, we use GENSIM [8] to train our models which are skip-gram and CBOW (Continuous Bag of Words). The detailed implementations are in their official website, and it won't be introduced here.

[8] https://radimrehurek.com/gensim/