Joint learning of Character and Word Embeddings

Authors
Xinxiong Chen, Lei Xu, Maosong Sun, Huanbo Luan

Presented by
Alabi Jesujoba, Kwabena Amponsah-Kaakyire

Outline

- Introduction
- CWE Model
  - Multiple-Prototype Character Embeddings
  - Word Selection for Learning
  - Initialization and Optimization
  - Complexity Analysis
- Experiments and Analysis
  - Datasets and Experiment Settings
  - Word Relatedness Computation
  - Analogical Reasoning
  - Influence of Learning Corpus Size
  - Case Study
- Related Work
- Conclusion and Future Work

Introduction

Word Embeddings

- Word representation aims at representing a word as a vector.
- The vector can be used to compute relatedness between words and also feed machine learning systems as features.
- Many NLP tasks conventionally take one-hot word representation,
  - by representing each word as the vocabulary-sized vector with one non-zero entry at the index of the occurrence of the word in the vocabulary.

Word Embeddings

- **The most critical flaw of one-hot representation:** not taking into account any semantic relatedness between words.
- Distributed word representation, also known as word embedding, was first proposed in (Rumelhart et al. , 1986)
- Word embedding encodes the semantic meanings of a word into a real-valued low-dimensional vector.

Word Embeddings: Issues

- ❖ In most embedding methods:
  - ➢ A word is taken as basic unit and
  - ➢ learn embeddings according to contexts ignoring internal structure of the words.
- ❖ Basically, they learn word embeddings according to external contexts of words in large-scale corpora.
- ❖ In Languages such as chinese, a word is composed of several characters and contains rich internal information.

Word Embeddings: Chinese example

- ❖ The composing characters of a word contain information about the meaning of the word
- ❖ Take a Chinese word “智能” pronounced “Zhìnéng” (intelligence) for example.
- ❖ The semantic meaning of the word “智能”
  - ➢ can be learned from its context in text corpora.
  - ➢ can also be inferred from the meanings of its characters “智” pronounced “Zhì” (intelligent) and “能” pronounced “néng” (ability).

Word Embeddings: Chinese example

- ❖ In the paper they took advantage of both internal characters and external contexts, and propose a new model for joint learning of character and word embeddings.
  - ➢ named as character-enhanced word embedding model (CWE).
  - ➢ Chinese was taken as an example for this work
- ❖ In CWE, both word and character embeddings are learned and maintained together.

CWE Model

CWE model: The Framework & Challenges

- The framework of CWE is a simple extension from other word embedding models.
- It faces several difficulties to consider characters into learning word embeddings.
- Compared with words,
  - Chinese characters are much more ambiguous (over 50,000 characters).
  - A character may play
    - different roles and have various semantic meanings in different words.
    - It will be insufficient to represent one character with only one vector.

CWE model: The Framework & Challenges

- Also, Not all Chinese words are semantically compositional, such as
    - transliterated words.

CWE model: The methods

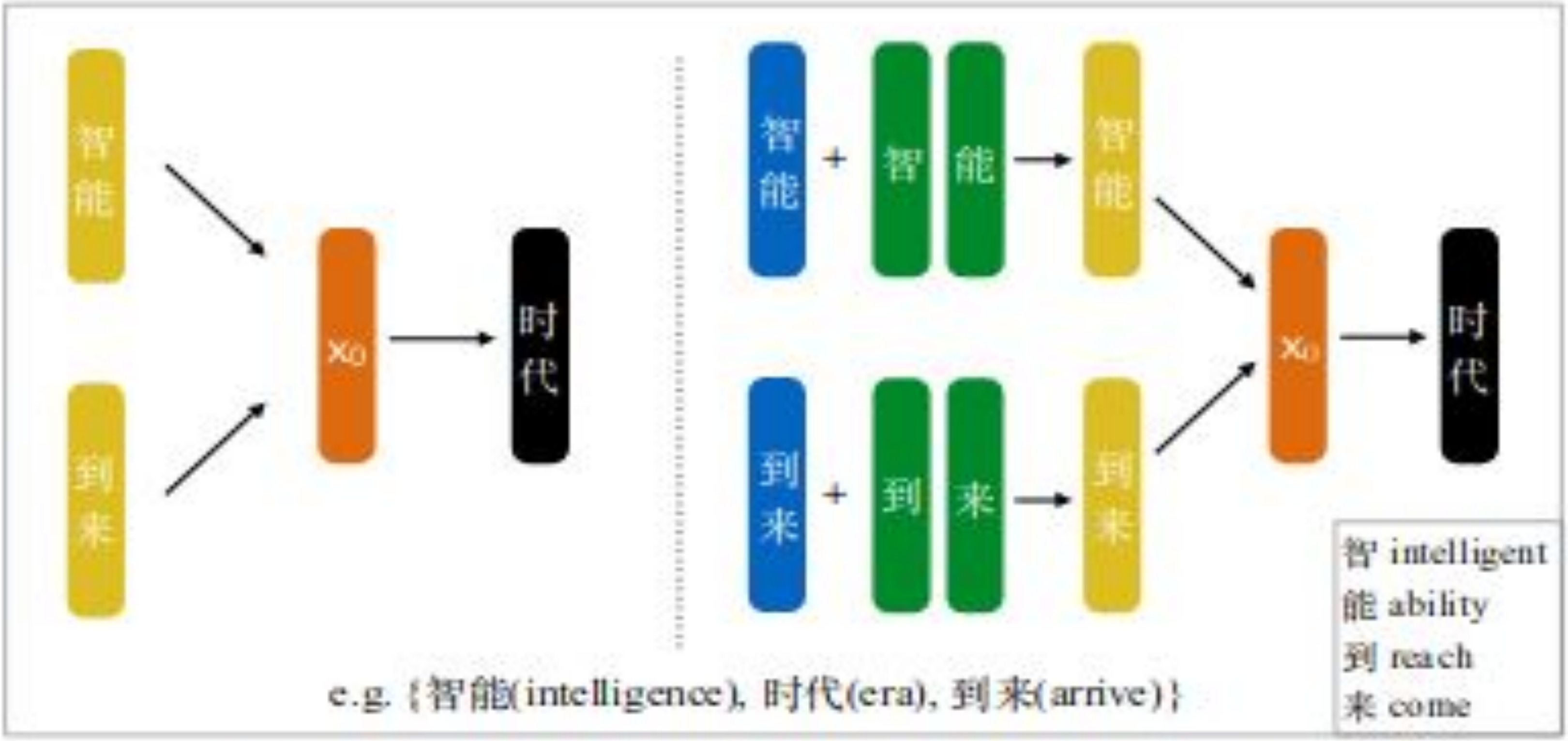
In the paper,

- ❖ They proposed multiple-prototype character embeddings.
  - ➢ obtain multiple vectors for a character, corresponding to various meanings of the character.
- ❖ They proposed several possible methods for multiple-prototype character embeddings:
  - ➢ Position-based
  - ➢ cluster-based and
  - ➢ nonparametric method.

CWE model: The methods

- They identified non-compositional words and built a wordlist in advance.
- Then they treated these words as a whole without considering their characters any more.

CBOW & CWE



(A) CBOW

(B)Character-enhanced Word Embedding

The Framework of the model: CBOW

CBOW aims at predicting the target word, given context words in a sliding window. Formally, given a word sequence D= {x1,...,xM} , the objective of CBOW is to maximize the average log probability

$$\mathcal{L}(D) = \frac{1}{M} \sum_{i=K}^{M-K} \log \Pr(x_i | x_{i-K}, \ldots, x_{i+K}). \qquad (1)$$

Here K is the context window size of a target word. CBOW formulates the probability Pr(xi|xi−K,...,xi+K) using a softmax function as follows

$$\Pr(x_i | x_{i-K}, \ldots, x_{i+K}) = \frac{\exp(\mathbf{x}_o^\top \cdot \mathbf{x}_i)}{\sum_{x'_i \in W} \exp(\mathbf{x}_o^\top \cdot \mathbf{x}'_i)}, \qquad (2)$$

Where W is the word vocabulary, xi is the vector representation of the target word xi, and xo is the average of all context word vectors

$$\mathbf{x}_o = \frac{1}{2K} \sum_{j=i-K,\ldots,i+K, j \neq i} \mathbf{x}_j. \qquad (3)$$

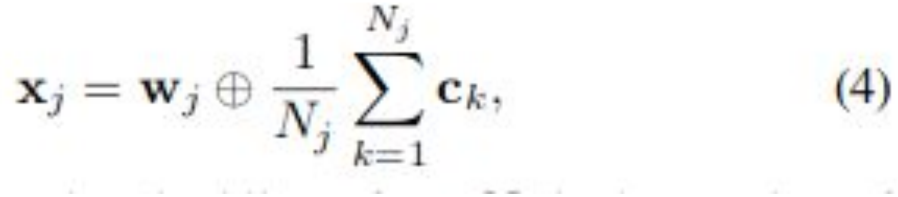
Character-Enhanced Word Embedding

CWE considers character embeddings in an effort to improve word embeddings. Let,

- C = the Chinese character set
- W = the Chinese word vocabulary as W.
- Ci = vector representing each character in the character set ci∈C,
- Wi = vector representing each word in the Vocabulary wi∈W.

- As we learn to maximize the average log probability in Equation(1) with a word sequence D={x1,.....xM}, context words are represented with both character embeddings and predict target words.

Xj is a context word



$$\mathbf{x}_j = \mathbf{w}_j \oplus \frac{1}{N_j} \sum_{k=1}^{N_j} \mathbf{c}_k, \qquad (4)$$

Character-Enhanced Word Embedding

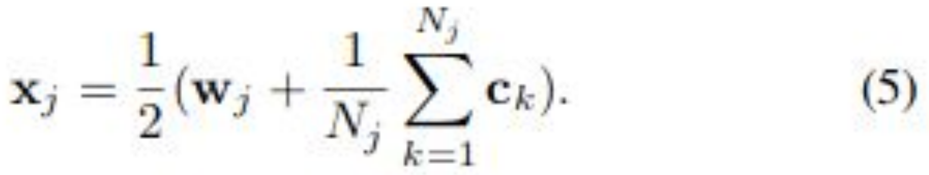
$$\mathbf{x}_j = \mathbf{w}_j \oplus \frac{1}{N_j} \sum_{k=1}^{N_j} \mathbf{c}_k, \qquad (4)$$

$$w_j = word\, embedding\, of\ x_j$$
$$N_j = number\ of\ characters\, in\, x_j$$
$$c_k = the\ embedding\, of\ the\, k - th\, character\ in\, xj$$
$$\bigoplus = composition\ operation$$

- ❖ There are two options for the composition operation:
  - ➢ addition (it requires $|w_j| = |c_k|$)
  - ➢ concatenation
- ❖ In experiment, concatenation (although more time consuming) does not outperform the addition operation significantly.

Character-Enhanced Word Embedding

Technically, they used



$$\mathbf{x}_j = \frac{1}{2}(\mathbf{w}_j + \frac{1}{N_j}\sum_{k=1}^{N_j}\mathbf{c}_k). \qquad (5)$$

Multiplying by ½ is crucial because it maintains similar length between embeddings of compositional and non-compositional words.

❖ The pivotal idea of CWE is to replace the stored vectors x in CBOW with real time compositions w and c, but shared same objective in Equation (1)

Multiple-Prototype Character Embeddings

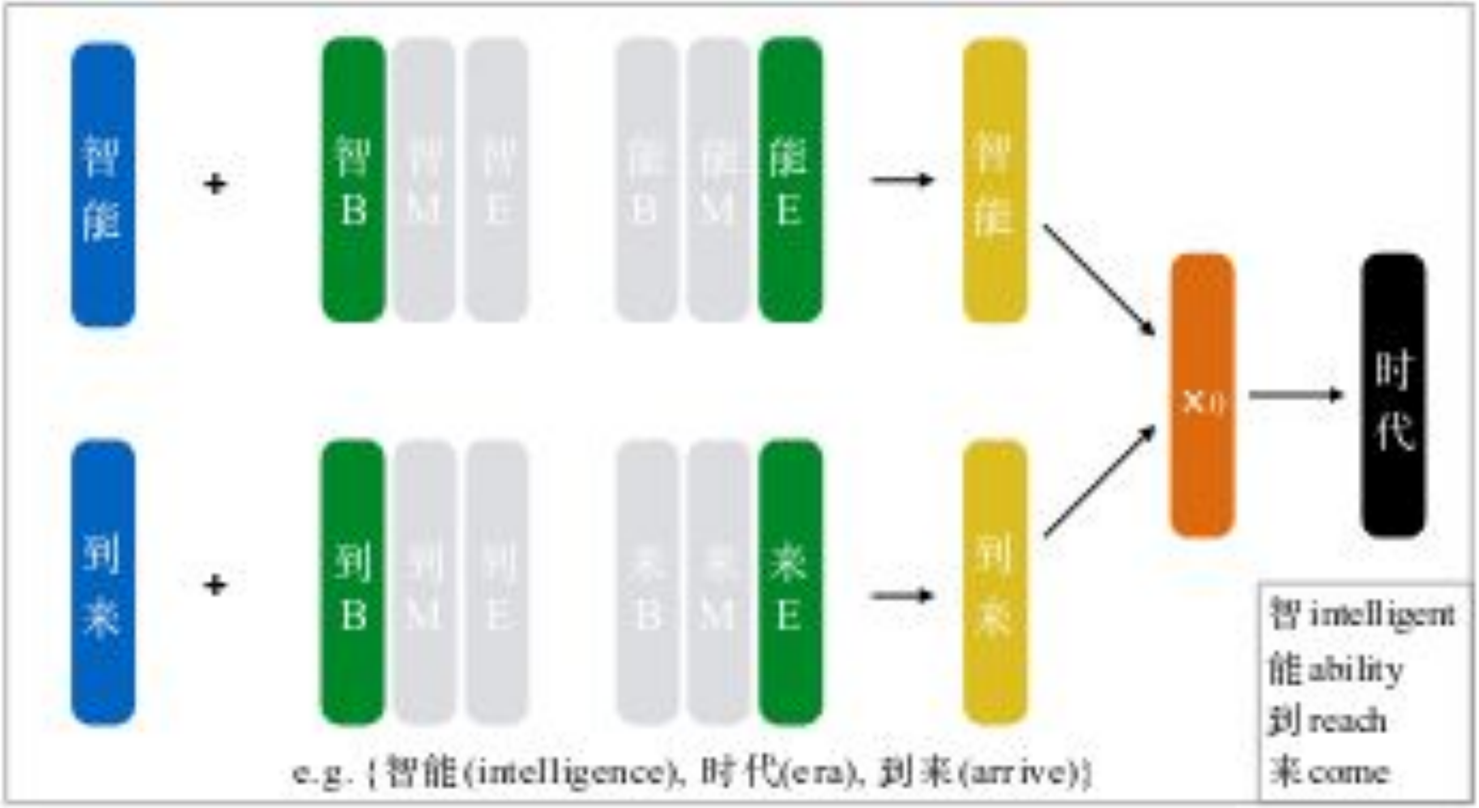
Chinese character are highly ambiguous. A multiple-prototype character embeddings to address this issue was proposed.

The idea: keep multiple vectors for one character, each corresponding to one of the meanings.

The several methods proposed for Multiple-prototype CE

- ❖ Position-based character embeddings
- ❖ Cluster-based character embeddings
- ❖ Nonparametric cluster-based character embeddings

Position-based Character Embeddings



Three embeddings were for each character c, (c^B, c^M, c^E) corresponding to its three types of positions in a word, i.e., Begin, Middle and End.

Position-based Character Embeddings

- Take a context word and its characters, $x_j = \{c_1, ...c_{N_j}\}$
- Take different embeddings of a character according to its position within x_j
- Embedding c_1^B is for the beginning character c_1 of the word x_j
- Embedding c_k^M is for the middle character $\{c_k | k = 2, ..., N_j - 1\}$
- Embedding $c_{N_j}^E$ for the last character c_{N_j}
- Equation (4) can be rewritten as

$$\mathbf{x}_j = \frac{1}{2}\Big(\mathbf{w}_j + \frac{1}{N_j}(\mathbf{c}_1^B + \sum_{k=2}^{N_j-1} \mathbf{c}_k^M + \mathbf{c}_{N_j}^E)\Big), \qquad (6)$$

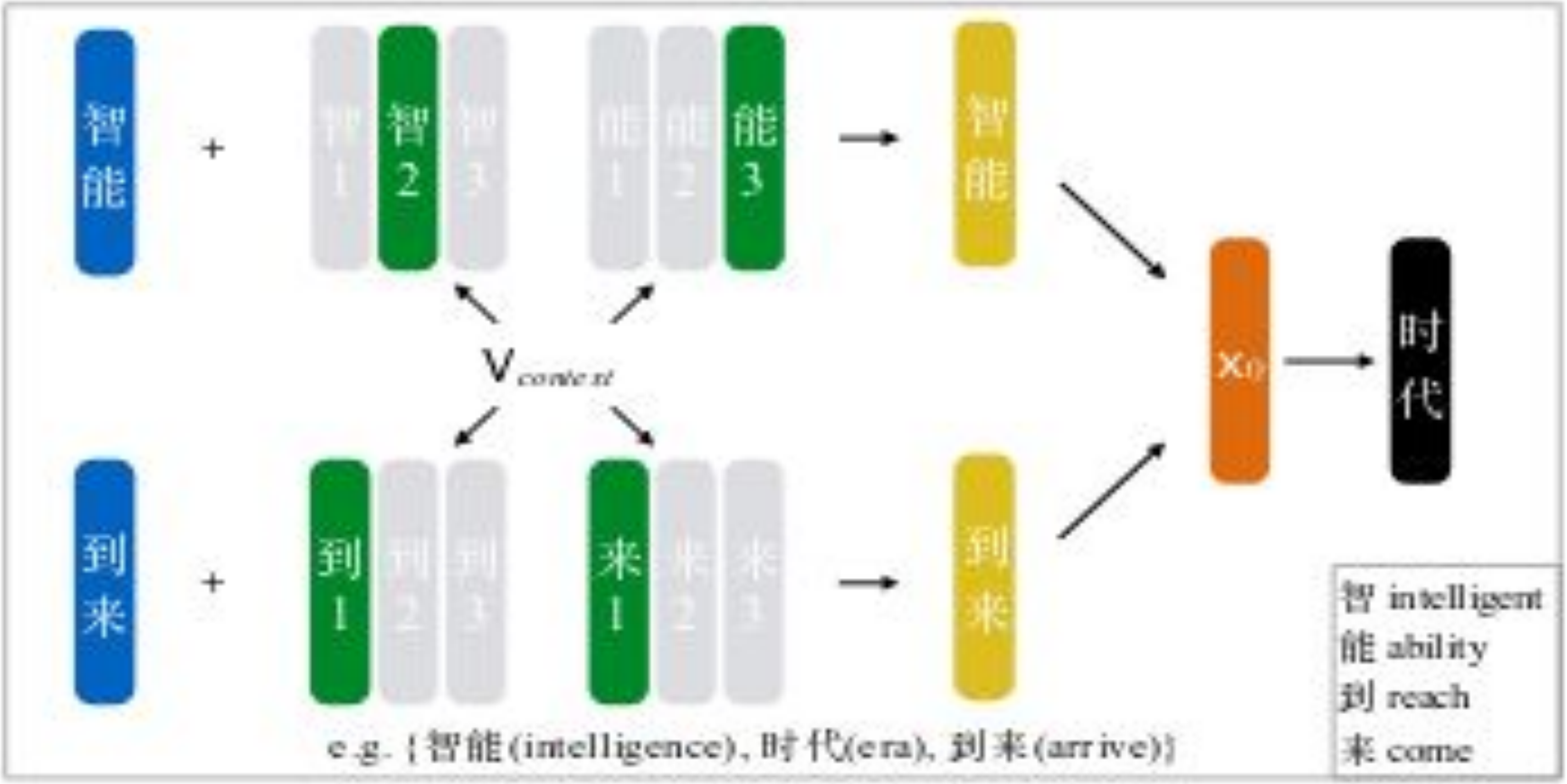
Position-based Character Embeddings

- ❖ In the position-based CWE, various embeddings of each are differentiated by the character position in the word.
- ❖ The embedding assignment for a specific character in a word can be automatically determined by it position.
- ❖ However, the exact meaning of a character is not only related to its position in a word.
- ❖ As a result, cluster-based character embeddings for CWE.

Cluster-based Character Embeddings

- It follows the method of multiple-prototype word embeddings(Huang et al., 2012)
- multiple representations to capture different senses and usages of a word
- The idea here is to simply cluster all occurrences of a character according to its context and form multiple prototypes of the N_c character.
- For each character c, we may cluster all its occurrences into
- Build one embedding for each cluster.

Cluster-based Character Embeddings



Cluster-based Character Embeddings

- ❖ take context word $x_j = \{c_1, ..., c_N\}$ for example, $\mathbf{c}_k^{r_k^{\max}}$ will be used to get x_j .

- ❖ Define S() as cosine similarity

$$r_k^{\max} = \arg\max_{r_k} S(\mathbf{c}_k^{r_k}, \mathbf{v}_{context}), \qquad (7)$$

where

$$\mathbf{v}_{context} = \sum_{t=j-K}^{j+K} \mathbf{x}_t = \sum_{t=j-K}^{j+K} \frac{1}{2}(\mathbf{w}_t + \frac{1}{N_t} \sum_{c_u \in x_t} \mathbf{c}_u^{\text{most}}). \qquad (8)$$

- ❖ c_u^{most} is the character embedding most frequently chosen by x_t in the previous training.

Cluster-based Character Embeddings

- ❖ After obtaining the optimal cluster assignment collection we $R = \left\{ r_1, ..., r_{N_j}^{max} \right\}$ can get the embedding $X_j \, of x_j \, as$

$$\mathbf{x}_j = \frac{1}{2}(\mathbf{w}_j + \frac{1}{N_j}\sum_{k=1}^{N_j} \mathbf{c}_k^{r_k^{max}}), \qquad (9)$$

- ❖ Note that, we can also apply the idea of clustering to position-based character embeddings.
- ❖ That is, for each position of a character (B,M,E), learn multiple embeddings to solve the possible ambiguity issue confronted in this position.
- ❖ This may be named as position-cluster-based character embedding

Nonparametric Cluster-based Character Embeddings

- ❖ The above hard cluster assignment is similar to the k-means clustering algorithm
- ❖ it learns a fixed number of clusters for each character.
- ❖ They proposed a nonparametric version of cluster-based character embeddings, which learns a varying number of clusters for each character.
- ❖ Following the idea of online nonparametric clustering algorithm (Neelakantan etal., 2014), the number of clusters for a character is unknown,and is learned during training.

N_{ck} *is the number of clusters associated with the character* c_k

- ❖ For the character c_k in a word x_j , the cluster assignment r_k

$$r_k = \begin{cases} N_{c_k} + 1, & \text{if } S(\mathbf{c}_k^{r_k}, \mathbf{v}_{context}) < \lambda \text{ for all } r_k. \\ r_k^{\max}, & \text{otherwise.} \end{cases} \qquad (10)$$

Word Selection for Learning

- ❖ There are many words in Chinese which do not exhibit semantic compositions from their characters.
- ❖ These words include:
  - ➢ Single-morpheme multi-character words, such as “琵琶” (lute), “徘徊” (wander), where these characters are hardly used in other words;
  - ➢ Transliterated words, such as“沙发” (sofa), “巧克力” (chocolate), which shows mainly phonetic compositions; and
  - ➢ Many entity names such as person names, location names and organisation names.

Word Selection for Learning

- To prevent the interference of non-compositional words, they proposed not to consider characters when learning these words,
- And learn both word and character embeddings for other words.
- A word list about transliterated words were manually built
- And they performed Chinese POS tagging to identify all entity names.
- Single-morpheme words almost do not influence modeling because their characters usually appear only in these words.

Initialization and Optimization

- Following the similar optimization scheme as that of CBOW used in (Mikolov et al., 2013), stochastic gradient descent (SGD) was used to optimize CWE models.
- Gradients are calculated using the back-propagation algorithm.
- Initialize both word and character embeddings atrandom like CBOW, Skip-Gram and GloVe.
- Initialization with pre-trained character embeddings may achieve a slightly better result.
- pre-trained character embeddings by simply regarding each character in the corpora as an individual word and learning character embeddings with word embedding models.

Computational complexity

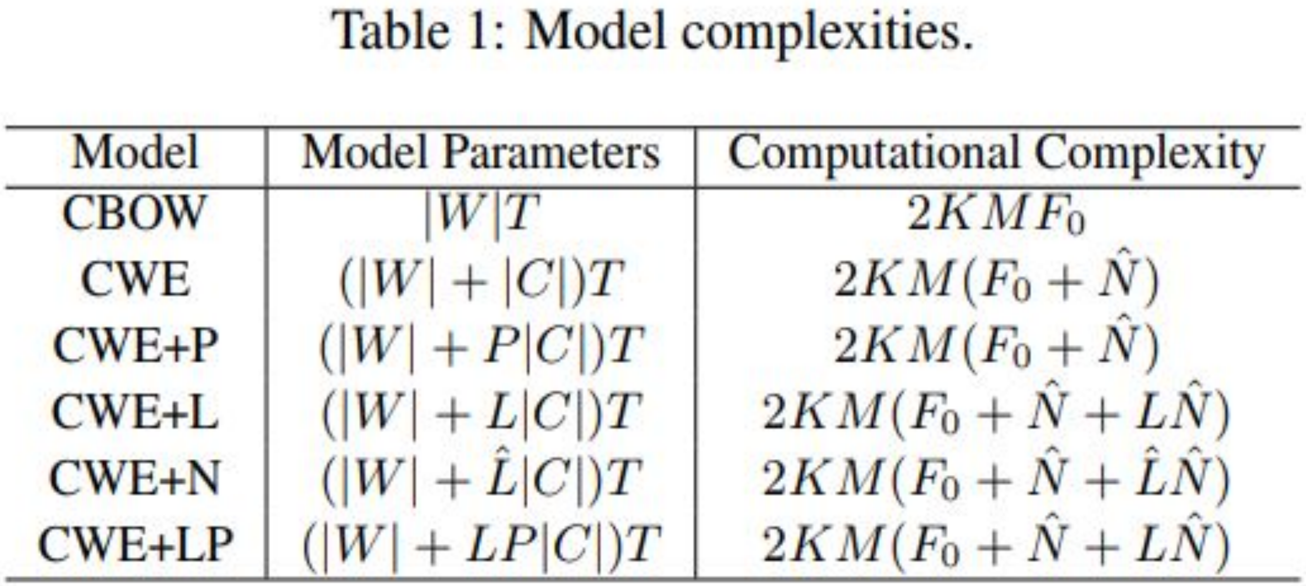


Table 1: Model complexities.

Model	Model Parameters	Computational Complexity
CBOW	$\|W\|T$	$2KMF_0$
CWE	$(\|W\| + \|C\|)T$	$2KM(F_0 + \hat{N})$
CWE+P	$(\|W\| + P\|C\|)T$	$2KM(F_0 + \hat{N})$
CWE+L	$(\|W\| + L\|C\|)T$	$2KM(F_0 + \hat{N} + L\hat{N})$
CWE+N	$(\|W\| + \hat{L}\|C\|)T$	$2KM(F_0 + \hat{N} + \hat{L}\hat{N})$
CWE+LP	$(\|W\| + LP\|C\|)T$	$2KM(F_0 + \hat{N} + L\hat{N})$

CWE+P - position-based character embeddings
CWE+L - cluster-based character embeddings
CWE+N - nonparametric cluster-based character embeddings
CWE+LP - position-cluster-based character embeddings

T - dimension of representation vectors
W - word vocabulary
C - character vocabulary size
P - number of character positions
L - number of clusters for each character
L^ - average number of nonparametric clusters for each character

2K - CBOW window size
M - corpus size
N^ - average number of characters of each word
F_0 - computational complexity of negative sampling and hierarchical softmax for each target word

Experiments and Analysis

Datasets and Experiment Settings

- A human-annotated corpus - news articles from The People’s Daily for embedding learning.
- 31 million words. Vocabulary size - 105000
- Character vocabulary size is 6000 (covering 96% characters in national standard charset GB2312).
- Vector dimension - 200
- Context window size as 5.
- Optimization - hierarchical softmax and 10-word negative sampling.
- Both word selection for CWE and use of pre-trained character embeddings.
- Baseline methods - CBOW, Skip-Gram and GloVe (same vector dimension and default parameters)
- Effectiveness of CWE evaluated on word relatedness computation and analogical reasoning.

Word Relatedness Computation

- Task: Each model is required to compute semantic relatedness of given word pairs.
- model performance: The correlations between results of models and human judgements.
- Datasets, wordsim-240 and wordsim-296.
- Wordsim-240 - 240 pairs of Chinese words and human-labeled relatedness scores. 233 word pairs in learning corpus .7 unseen word pairs.
- Wordsim-296 - 296 pairs of Chinese words and human-labeled relatedness scores. 280 word pairs in learning corpus .16 unseen word pairs.

Word Relatedness Computation

- Similarity measure:
  - Spearman correlation (ρ) for relatedness scores between a model and the human judgements.
  - Cosine similarity for relatedness scores between CWE and other baseline embedding methods
  - “For a word pair with new words, we assume its similarity is 0 in baseline methods since we can do nothing more, while CWE can generate embeddings for these new words from their character embeddings for relatedness computation.”

Word Relatedness Computation

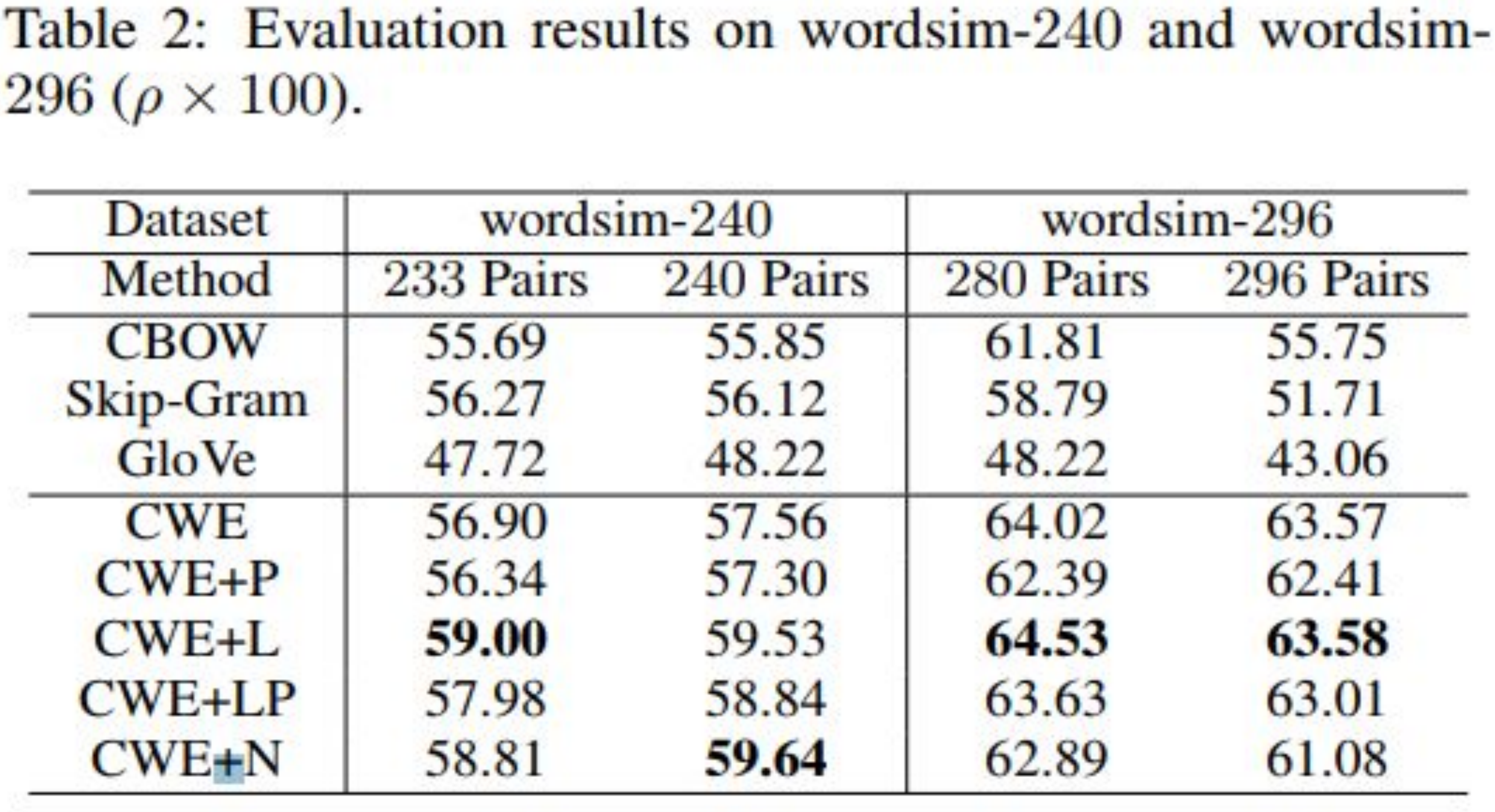


Table 2: Evaluation results on wordsim-240 and wordsim-296 ($\rho \times 100$).

Dataset	wordsim-240		wordsim-296	
Method	233 Pairs	240 Pairs	280 Pairs	296 Pairs
CBOW	55.69	55.85	61.81	55.75
Skip-Gram	56.27	56.12	58.79	51.71
GloVe	47.72	48.22	48.22	43.06
CWE	56.90	57.56	64.02	63.57
CWE+P	56.34	57.30	62.39	62.41
CWE+L	**59.00**	59.53	**64.53**	**63.58**
CWE+LP	57.98	58.84	63.63	63.01
CWE+N	58.81	**59.64**	62.89	61.08

Word Relatedness Computation

- ❖ Cluster-based extensions including +P, +LP and +N perform better than CWE
  - ➢ indicates that senses of characters is important for character embeddings and position information is not adequate in addressing ambiguity.
- ❖ The addition of 7 word pairs with new words does not cause significant change of correlations for both baselines and CWE methods.
  - ➢ Reason: the 7 word pairs are mostly unrelated. The default setting of 0 in baseline methods is basically consistent with the fact.

Word Relatedness Computation

- ❖ For wordsim-296, the performance of baseline methods drop dramatically when adding 16 word pairs of new words,
- ❖ while the performance of CWE and its extensions remain stable.
- ❖ The reason is that the baseline methods cannot handle these new words appropriately.
- ❖ For example, “老虎” (tiger) and “美洲虎” (jaguar) are semantically relevant, but the relatedness is set to 0 in baseline methods simply because “美洲虎” does not appear in the corpus, resulting in all baseline methods ranking the word pair much lower than where it should be.

Word Relatedness Computation

- In contrast, CWE and its extensions compute the semantic relatedness of these word pairs much closer to human judgements.

- Common to a new word in Chinese than a new character

- CWE can easily cover all Chinese characters in these new words and provide useful information about their semantic meanings for computing the relatedness.

Word Relatedness Computation

- Side effect: CWE methods will tend to misjudge the relatedness of two words with common characters.

- For example, the relatedness of word pair “肥皂剧” (soap opera) and “歌 剧” (opera) and the word pair “电话” (telephone) and “回话” (reply) are overestimated due to having common characters (i.e., “剧” and “话”, respectively).

Analogical Reasoning

- Task: word analogies
- Eg. “ 男人 (man) : 女 人 (woman) :: 父亲 (father) : ?”.
  - Embedding methods are expected to find a word x such that its vector x is closest to vec(女人) - vec(男人) + vec(父亲) according to the cosine similarity.
  - Model answer is considered correct if the word “母亲” (mother) is found.

Analogical Reasoning

- No existing Chinese analogical reasoning dataset
- Manually built a Chinese dataset consisting of
  - 1, 125 analogies of 3 analogy types:
    - capitals of countries (687 groups);
    - states/provinces of cities (175 groups);
    - family words (240 groups).
  - Training corpus covers more than 97% of all the test corpus.

Analogical Reasoning

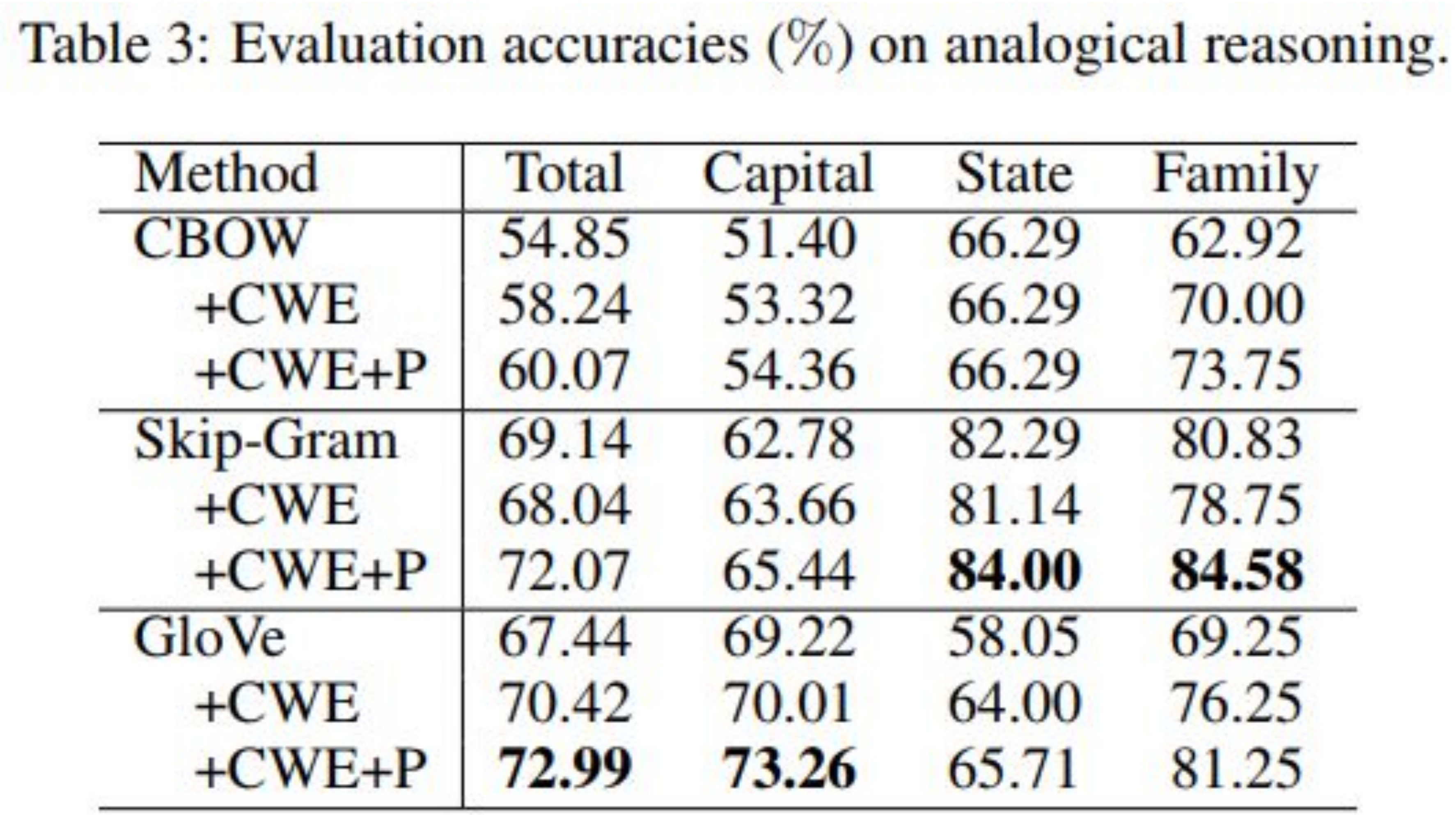


Table 3: Evaluation accuracies (%) on analogical reasoning.

Method	Total	Capital	State	Family
CBOW	54.85	51.40	66.29	62.92
+CWE	58.24	53.32	66.29	70.00
+CWE+P	60.07	54.36	66.29	73.75
Skip-Gram	69.14	62.78	82.29	80.83
+CWE	68.04	63.66	81.14	78.75
+CWE+P	72.07	65.44	**84.00**	**84.58**
GloVe	67.44	69.22	58.05	69.25
+CWE	70.42	70.01	64.00	76.25
+CWE+P	**72.99**	**73.26**	65.71	81.25

Analogical Reasoning

- For CBOW, SkipGram and GloVe, most of their CWE versions consistently outperform the original model.
- Indicates necessity of considering character embeddings for word embeddings.
- CWE models can improve the embedding quality of all words, not only those words whose characters are considered for learning.
- For example, in the type of capitals of countries, all the words are entity names whose characters are not used for learning. CWE model can still make an improvement on this type as compared to baseline models.

Analogical Reasoning

- As reported in [Mikolov et al., 2013; Pennington et al., 2014], Skip-Gram and GloVe perform better on analogical reasoning than CBOW.
- By integrating the idea of CWE to Skip-Gram and GloVe, encouraging increase of 3% to 5% is achieved.
- This indicates the generality of effectiveness of CWE.

Influence of Learning Corpus Size

Task: Investigate the influence of corpus size for word embeddings using the word relatedness computation task using as an example.

Influence of Learning Corpus Size

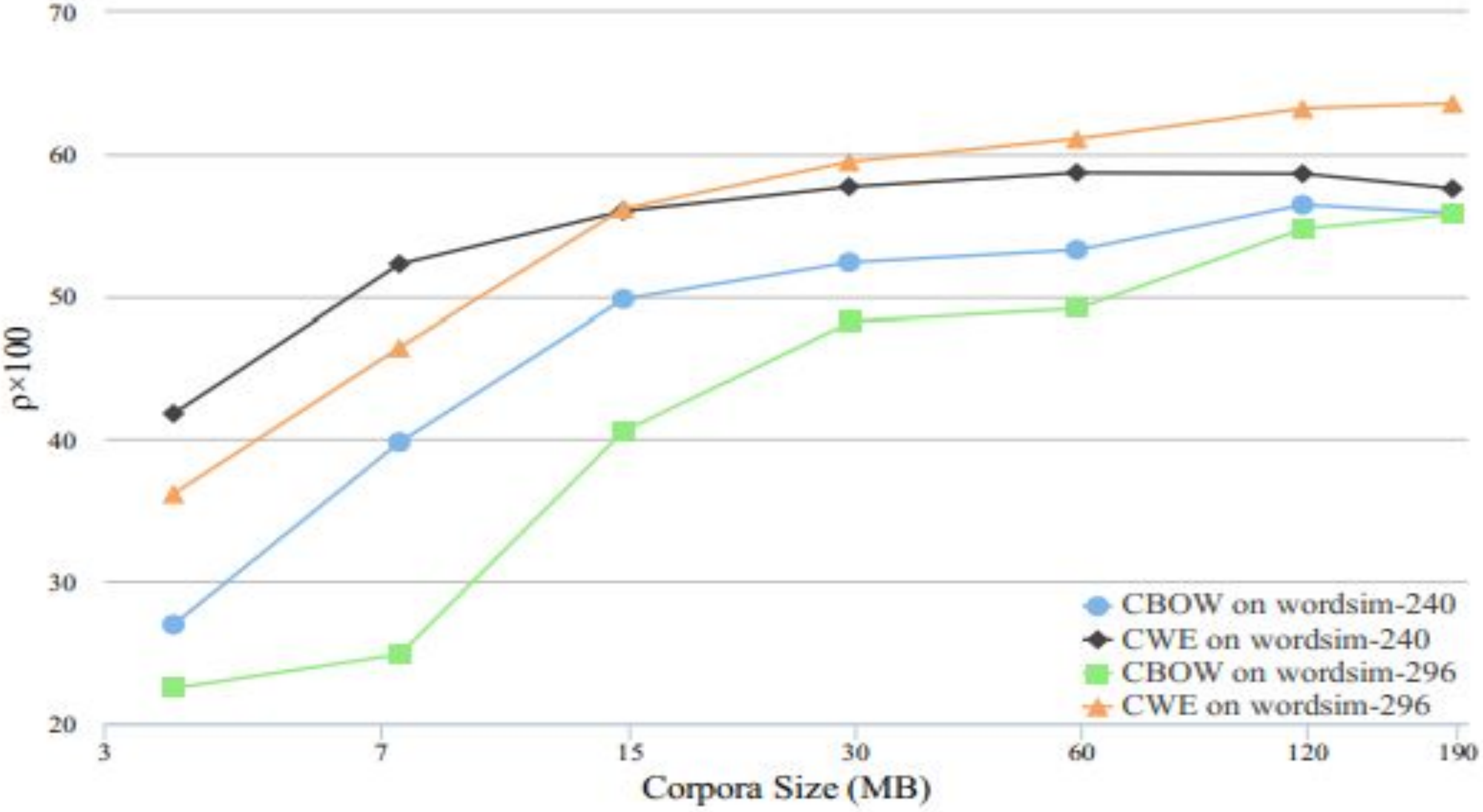


Figure 4: Results on wordsim task with different corpora size.

Case Study

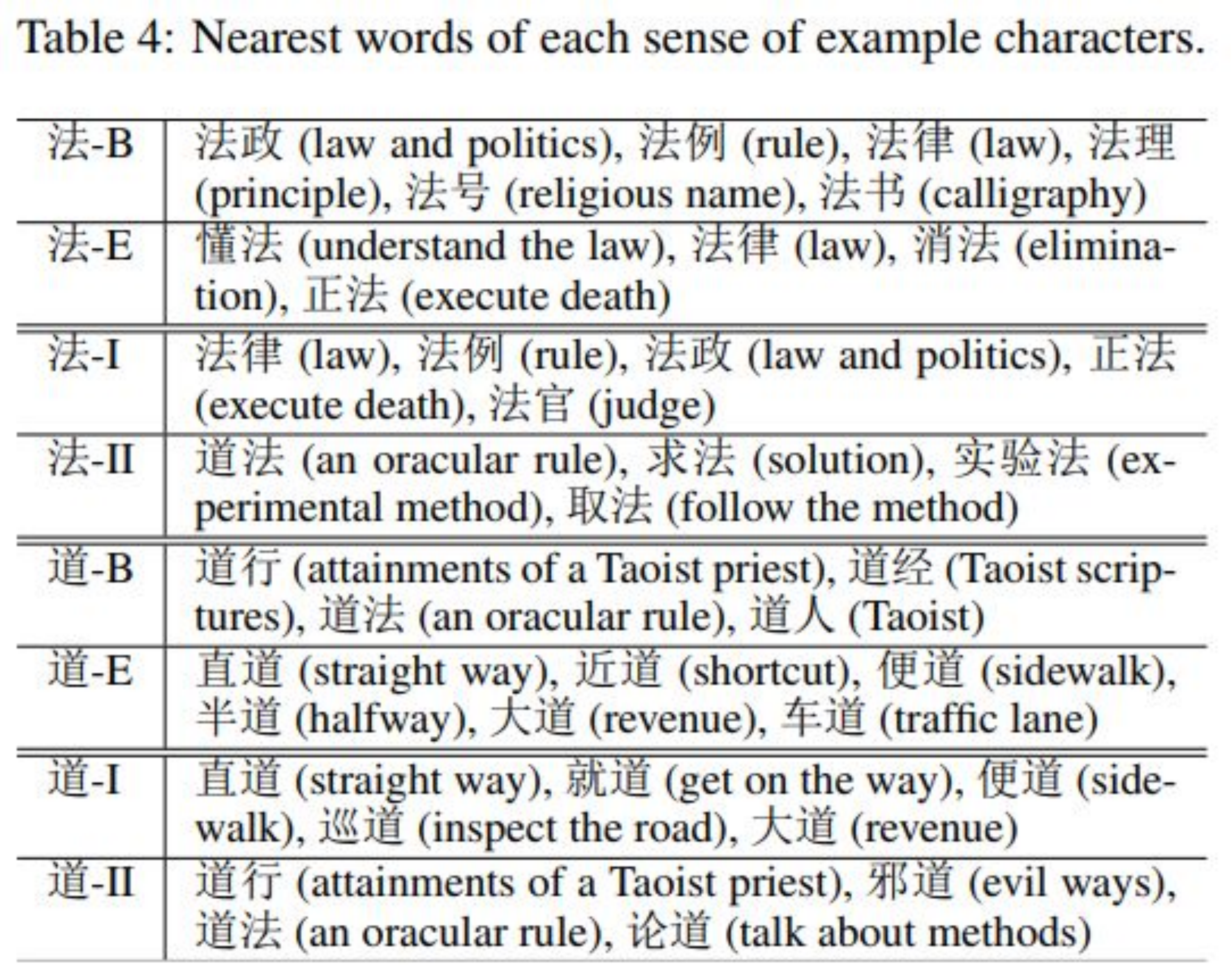


Table 4: Nearest words of each sense of each example characters.

法-B	法政 (law and politics), 法例 (rule), 法律 (law), 法理 (principle), 法号 (religious name), 法书 (calligraphy)
法-E	懂法 (understand the law), 法律 (law), 消法 (elimination), 正法 (execute death)
法-I	法律 (law), 法例 (rule), 法政 (law and politics), 正法 (execute death), 法官 (judge)
法-II	道法 (an oracular rule), 求法 (solution), 实验法 (experimental method), 取法 (follow the method)
道-B	道行 (attainments of a Taoist priest), 道经 (Taoist scriptures), 道法 (an oracular rule), 道人 (Taoist)
道-E	直道 (straight way), 近道 (shortcut), 便道 (sidewalk), 半道 (halfway), 大道 (revenue), 车道 (traffic lane)
道-I	直道 (straight way), 就道 (get on the way), 便道 (sidewalk), 巡道 (inspect the road), 大道 (revenue)
道-II	道行 (attainments of a Taoist priest), 邪道 (evil ways), 道法 (an oracular rule), 论道 (talk about methods)

Related Work, Future Work and Conclusion

Related Work

- ❖ [Collobert et al., 2011] - extra features such as capitalization to enhance word vectors.
  - ➢ **cannot generate high-quality word embeddings for rare words.**
- ❖ Reveal morphological compositionality.
  - ➢ [Alexandrescu and Kirchhoff, 2006] - a factored neural language model where each word is viewed as a vector of factors.
  - ➢ [Lazaridou et al., 2013] - the application of compositional distributional semantic models, originally designed to learn phrase meanings, for derivational morphology.
  - ➢ [Luong et al., 2013] proposed a recursive neural network (RNN) to model morphological structure of words.
  - ➢ [Botha and Blunsom, 2014] proposed a scalable method for integrating compositional morphological representations into a log-bilinear language model.
  - ➢ **Mostly sophisticated and task-specific, which make them non-trivial to be applied to other scenarios.**

Related Work

- CWE is a simple and general way to integrate the internal knowledge (character) and external knowledge (context) to learn word embeddings,
  - capable of being extended in various models and tasks.
- [Huang et al., 2012] proposed a method of multiple embeddings per word to resolve ambiguity.
- Little work has addressed the ambiguity issue of characters or morphemes, which is the crucial challenge when dealing with Chinese characters.
  - CWE provides an effective and efficient solution to character ambiguity.
- Although this paper focuses on Chinese, our model deserves to be applied to other languages, such as English where affixes may have various meanings in different words.

Future Work

- An addition operation for semantic composition between word and character embeddings.
  - Motivated by recent works on semantic composition models based on matrices or tensors
  - May explore more sophisticated composition models to build word embeddings from character embeddings to endorse CWE with more powerful capacity of encoding internal character information.
- CWE may learn to assign various weights for characters within a word.
- Explore rich information about words to build a word classifier for selection.

Conclusion

❖ CWE Introduces internal character information into word embedding methods to alleviate excessive reliance on external information.

❖ Can be easily integrated into existing word embedding models including CBOW, Skip-Gram and GloVe.

❖ Has been shown to consistently and significantly improve the quality of word embeddings.

References

https://www.aclweb.org/anthology/D17-1027

https://arxiv.org/abs/1504.06654

https://www.aclweb.org/anthology/P12-1092

https://www.ijcai.org/Proceedings/15/Papers/178.pdf