

Comparing Static and Contextual Distributional Semantic Models on Intrinsic Tasks: An Evaluation on Mandarin Chinese Datasets

Pranav A, Yan Cong, Emmanuele Chersoni, Yu-Yin Hsu, Alessandro Lenci

Can skipgram outperform transformer models in classical distributional tasks? Does tokenization matter?

What is distributional semantics

- Distributional Semantics: assumes that words appearing in similar contexts have similar meanings.
- Distributional Semantic Models (DSMs) are data-driven semantic representations from large text corpora.
- Static models: a single, global semantic representation for each word type (word2vec, skipgram)
- Dynamic models: Contextualized (like ELMO, BERT, GPT)

Chinese DSM

Chinese: definition of words is not trivial

Characters are fundamental units of Chinese

Chinese Transformers are tokenized by characters.

We want to assess how these character based transformers behave when it comes to classical distributional tasks which are word-level.

Static Models

Using embeddings by Li et al., based on skipgram.

We use different context features like words, characters, n-grams.

Model	Context Features
Skip Gram	Words
Skip Gram N	Words, Ngrams
Skip Gram C	Words, Characters
Skip Gram N+C	Words, Ngrams, Characters

Chinese Word Vectors 中文词向量

[中文](#)

This project provides 100+ Chinese Word Vectors (embeddings) trained with different **representations** (dense and sparse), **context features** (word, ngram, character, and more), and **corpora**. One can easily obtain pre-trained vectors with different properties and use them for downstream tasks.

Moreover, we provide a Chinese analogical reasoning dataset **CA8** and an evaluation toolkit for users to evaluate the quality of their word vectors.

Contextualized Models

1. We sample n sentences (10-100) that contain the target dataset words.
2. We extract contextualized vectors of each sampled sentence.
3. We average them out to form single type-level vector.

We use these transformers for our experiments:

1. BERT Chinese (base and large)
2. GPT-2 Chinese
3. DeBERTa Chinese

Layer wise assessment

We extracted representations from different sets of layers to understand how different layers perform in our task.

- First 4 layers
- Middle 4 layers
- Last 4 layers
- Final output layer

Benchmarks

We chose five intrinsic tasks:

- Similarity Estimation
- Word Associations
- Semantic Analogies,
- Identification of Semantic Relations
- Semantic Clustering

Similarity estimation benchmarks

COS960 dataset, 960 word pairs

Word Pair	Translation	Score
共存 - 溶合	co-exist - integrate	0.6
窥视 - 窥探	peak-observe - peak-detect	3.86
船头 - 船尾	ship-head - ship-tail	0.93

Table 2: Examples of word pairs and average human scores in the COS960 dataset.

Similarity estimation benchmarks

Skipgram models with n-grams and characters are on par with contextualized models.

Adding extra context helps in skipgram performance.

Model	Spearman		Cosine	
	ρ	r	ρ	r
BERT-Base First 4	0.73	0.70	0.71	0.68
BERT-Base Last	0.74	0.70	0.74	0.69
BERT-Base Last 4	0.73	0.69	0.73	0.67
BERT-Base Middle 4	0.72	0.69	0.72	0.67
DeBERTa First 4	0.74	0.72	0.73	0.69
DeBERTa Last	0.74	0.71	0.72	0.66
DeBERTa Last 4	0.74	0.72	0.73	0.67
DeBERTa Middle 4	0.73	0.70	0.74	0.70
GPT-2 First 4	0.73	0.70	0.70	0.66
GPT-2 Last	0.70	0.67	0.44	0.37
GPT-2 Last 4	0.71	0.68	0.61	0.56
GPT-2 Middle 4	0.72	0.69	0.71	0.66
SkipGram N+C	0.75	0.70	0.75	0.70
SkipGram N	0.69	0.64	0.69	0.64
SkipGram C	0.71	0.67	0.71	0.67
SkipGram	0.65	0.60	0.65	0.61

Word Associations Task

Using FAST-zh dataset, 300 tuples of 4 words.

We designed 2 settings for this task:

- Multiple choice: Find out the first associate in the tuple (accuracy)
- Retrieval: Find the first associate within all first associates (mean rank)

Stimulus	First	Higher	Random
活 (live)	死 (die)	人生 (life)	人才 (talent, talented person)

Table 3: Example of a tuple from the FAST-Zh dataset.

Word association task

Static models clearly outperform contextualized models.

Not much difference between the layers' performances.

In general, the model goes for first associate for nearly 70% of the time and goes for a higher associate 30% of the time.

Model	Accuracy		Mean Rank	
	cos	ρ	cos	ρ
BERT-Base First 4	0.68	0.68	2.60	2.61
BERT-Base Last	0.69	0.69	2.28	2.42
BERT-Base Last 4	0.71	0.71	2.40	2.25
BERT-Base Middle 4	0.70	0.71	2.41	2.32
DeBERTa First 4	0.69	0.68	2.56	2.49
DeBERTa Last	0.71	0.70	2.40	2.28
DeBERTa Last 4	0.69	0.72	2.41	2.25
DeBERTa Middle 4	0.68	0.69	2.42	2.28
GPT-2 First 4	0.68	0.66	2.61	2.60
GPT-2 Last	0.58	0.68	2.87	2.42
GPT-2 Last 4	0.66	0.68	2.53	2.39
GPT-2 Middle 4	0.66	0.66	2.54	2.53
SkipGram	0.73	0.73	2.16	2.17
SkipGram C	0.72	0.71	2.10	2.12
SkipGram N	0.74	0.73	2.10	2.13
SkipGram N+C	0.73	0.72	2.19	2.20

Table 9: Word Associations results on the FAST-zh dataset. We show accuracy (the higher the better) and mean rank (the lower the better) using cosine (*cos*) and Spearman (ρ) as similarity metrics.

Semantic Analogies Task

Using CA8 dataset, 7353 pairs of analogies.

The task is to find out correct target word within the vocabulary of the dataset words (reported in accuracy).

Analogy	Target Word
中国: 北京 = 意大利:??? (China:Beijing=Italy:???)	罗马 (Rome)

Table 4: Example of a semantic analogy for the country-capital relation from the CA8 dataset.

Semantic Analogies

Skip Gram models achieve much higher scores compared to the contextualized models, showing near-perfect performance across all settings.

Later layers in transformers tend to perform better than earlier layers in this task

Model	cos	ρ
BERT-Base First 4	0.39	0.39
BERT-Base Last	0.84	0.84
BERT-Base Last 4	0.82	0.82
BERT-Base Middle 4	0.67	0.68
DeBERTa First 4	0.44	0.44
DeBERTa Last	0.63	0.63
DeBERTa Last 4	0.63	0.63
DeBERTa Middle 4	0.57	0.57
GPT-2 First 4	0.41	0.41
GPT-2 Last	0.38	0.38
GPT-2 Last 4	0.44	0.43
GPT-2 Middle 4	0.44	0.43
SkipGram N+C	0.93	0.93
SkipGram N	0.97	0.97
SkipGram C	0.91	0.91
SkipGram	0.93	0.92

Table 11: Semantic analogies results for the CA8 dataset. We show accuracy using cosine (cos) and Spearman (ρ) as similarity metrics.

Semantic Relations Task

Using EVALution-MAN dataset.

3, 923 word pairs, covering the relations of synonymy, hyponymy and antonymy.

Word Pair	Translation	Rel.
不僅僅 - 不單單	not only - not just	syno.
海獅 - 水中生物	sea lion - marine animals	hyper.
男性 - 女性	male - female	anto.
新光 - 勸告	new light - advice	random

Table 5: An example of semantic *relata* for each relation in EVALution-MAN dataset.

Semantic Relations Task

Two tasks:

- Finding out the related pairs. Here we group related words (synonyms, antonyms and hyponyms) and unrelated words (random). Based on vector similarity score we rank them and report the results in average precision.
- Finding out synonyms. Here we group synonyms and other words (antonyms, hyponyms and random). Based on vector similarity score we rank them and report the results in average precision.

Semantic Relations

Contextualized models, particularly BERT, consistently outperform all the competitors by a large margin for all metrics and layer setting.

Static models, while not as powerful as contextualized models, still exhibit strong performance.

Model	cos		ρ	
	Rel.	Syn.	Rel.	Syn.
BERT-Base First 4	0.95	0.50	0.92	0.52
BERT-Base Last	0.93	0.56	0.94	0.61
BERT-Base Last 4	0.96	0.58	0.94	0.61
BERT-Base Middle 4	0.96	0.59	0.94	0.61
DeBERTa First 4	0.69	0.25	0.83	0.42
DeBERTa Last	0.76	0.33	0.91	0.56
DeBERTa Last 4	0.72	0.29	0.89	0.54
DeBERTa Middle 4	0.71	0.29	0.87	0.52
GPT-2 First 4	0.89	0.49	0.91	0.53
GPT-2 Last	0.78	0.44	0.92	0.59
GPT-2 Last 4	0.87	0.49	0.92	0.59
GPT-2 Middle 4	0.91	0.52	0.91	0.57
SkipGram	0.80	0.26	0.80	0.25
SkipGram C	0.82	0.33	0.81	0.32
SkipGram N	0.79	0.25	0.79	0.25
SkipGram N+C	0.80	0.30	0.79	0.29

Table 12: Semantic Relations results on EVALution-MAN. We show average precision using cosine (cos) and Spearman (ρ) as metrics on related (Rel.) and synonymy (Syn.) classes.

Semantic Clustering Task

Two tasks:

- Binder-zh: 535 words in 11 concept classes
- Zhong22: 664 words in 2 classes (abstract and concrete).

The evaluation metrics are homogeneity and completeness.

Type-POS	No. of items
Concrete Objects - Nouns	275
Living Things - Nouns	126
Other Natural Objects - Nouns	19
Artifacts - Nouns	130
Concrete Events - Nouns	60
Abstract Entities - Nouns	99
Concrete Actions - Verbs	52
Abstract Actions - Verbs	5
States - Verbs	5
Abstract Properties - Adjectives	13
Physical Properties - Adjectives	26

Table 6: Concept classes, parts-of-speech and number of words in the Binder-zh norms.

Semantic Clustering

Static models perform better in distinguishing abstract and concrete words in the Zhong22 dataset, with Skip Gram using only words as contexts.

Contextualized models, except for BERT and DeBERTa in middle-to-later layers, achieve lower scores.

However, for the fine-grained semantic distinctions of the Binder dataset, BERT and DeBERTa in the middle and late layers outperform static models.

Model	Zhong		Binder-zh	
	H	C	H	C
BERT-Base First 4	0.11	0.12	0.10	0.31
BERT-Base Last	0.28	0.31	0.14	0.37
BERT-Base Last 4	0.16	0.24	0.12	0.35
BERT-Base Middle 4	0.25	0.28	0.12	0.34
DeBERTa First 4	0.09	0.16	0.13	0.37
DeBERTa Last	0.21	0.23	0.13	0.36
DeBERTa Last 4	0.29	0.32	0.14	0.36
DeBERTa Middle 4	0.29	0.29	0.17	0.46
GPT-2 First 4	0.07	0.07	0.11	0.33
GPT-2 Last	0.03	0.03	0.04	0.07
GPT-2 Last 4	0.04	0.04	0.11	0.28
GPT-2 Middle 4	0.07	0.11	0.15	0.40
SkipGram N+C	0.20	0.25	0.08	0.22
SkipGram N	0.21	0.25	0.07	0.20
SkipGram C	0.34	0.35	0.03	0.21
SkipGram	0.37	0.37	0.03	0.21

Table 13: Semantic Clustering results on Zhong22 and Binder-zh. We show homogeneity (H) and completeness (C) using agglomerative clustering.

Conclusions

We present a study of static vs contextualized embeddings on distributional tasks.

Our results reveal that static models remain stronger for some of the classical tasks that consider word meaning independent of context.

Contextualized models excel in identifying semantic relations between word pairs and in the categorization of words into abstract semantic classes, which probably benefit from context-specific semantic cues

Transformers' vocab is made of characters, it does not have native representation of words. A reason why they tend to underperform in most of these distributional tasks.

More work needs to be done on other languages on understand how tokenization might affect the performance on intrinsic benchmarks.