Syntactic Dependencies and Distributed Word Representations for Chinese Analogy Detection and Mining

Likun Qiu[1,2], **Yue Zhang**[2], **Yanan Lu**[3]
[1]School of Chinese Language and Literature, Ludong University, China
[2]Singapore University of Technology and Design, Singapore
[3]Computer School, Wuhan University, China
qiulikun@pku.edu.cn, yue_zhang@sutd.edu.sg, luyanan@whu.edu.cn

Abstract

Distributed word representations capture relational similarities by means of vector arithmetics, giving high accuracies on analogy detection. We empirically investigate the use of syntactic dependencies on improving Chinese analogy detection based on distributed word representations, showing that a dependency-based embeddings does not perform better than an ngram-based embeddings, but dependency structures can be used to improve analogy detection by filtering candidates. In addition, we show that distributed representations of dependency structure can be used for measuring relational similarities, thereby help analogy mining.

1 Introduction

Relational similarity measures the correspondence between word-word relations (Medin et al., 1990). It is relevant to many tasks in NLP (Turney, 2006), such as word sense disambiguation, information extraction, question answering, information retrieval, semantic role identification and metaphor detection. Typical tasks on relational similarity include *analogy detection*, which measures the degree of relational similarities, and *analogy mining*, which extracts analogous word pairs from unstructured text.

Recently, distributed word representations (i.e. *embeddings*) (Mikolov et al., 2013a; Mikolov et al., 2013b; Levy and Goldberg, 2014b) have been used for unsupervised analogy detection. Mikolov et al. use attributional similarities between words in a relation to compute relational similarities, and show that the method outperforms the best system in the SemEval 2012 shared task on analogy detection. Levy and Goldberg (2014b) further improve Mikolov's relational similarity measure method using novel arithmetic combinations of attributional similarities. For simplicity, we call the method of Mikolov et al. *embedding-based analogy detection*, without stressing the difference between *distributed* and *distributional* (i.e. counting-based) word representations.

Most work on embedding-based analogy detection uses relational similarities as a measure of the quality of embeddings. However, relatively little has been done in the opposite direction, exploring how to leverage embeddings for improving relational similarity algorithms. We empirically study the use of word embeddings for Chinese analogy detection and mining, leveraging syntactic dependencies, which has been shown to be closely associated with semantic relations (Levin, 1993; Chiu et al., 2007). Compared with many other languages, this association is particularly strong for Chinese, which is fully configurational and lacks morphology. To our knowledge, relatively little work has been reported on Chinese relational similarities, compared to other tasks in Chinese NLP, including syntactic parsing, information extraction and machine translation.

We work on three specific problems. First, we study the effect of dependency-based word embeddings for analogy detection. There are two variations of Mikolov et al's *skip-gram* embedding model, one training the distributed word representation of a word using its context words in local ngram window (Mikolov et al., 2013a), and the other training the distributed representation of a word using words in a syntactic dependency context (Levy and Goldberg, 2014b; Bansal et al., 2014). The latter has attracted much recent atten-

Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pages 2441–2450,
Lisbon, Portugal, 17-21 September 2015.

tion due to its potential in capturing more syntactic regularities. It has been shown to outperform the former in a variety of NLP tasks, and can potentially also improve relation similarity. Our experiments on both English and Chinese show that the dependency-context embeddings consistently under-perform ngram-context embeddings. We give some theoretical justifications to the findings.

Second, we propose to use syntactic dependencies as a context for improving embedding-based analogy detection, pruning the search space and filtering noise using syntactic dependencies. While highly useful for measuring relational similarities, attributional similarities between words are not the only source of information for analogy detection. Traditional methods, such as Turney and Littman (2005), Turney (2006), Chiu et al. (2007) and Ó Séaghdha and Copestake (2009), also leverage context between word pairs in a corpus for better accuracies, which the current embedding-based methods ignore. Results show that our proposed method achieves significant improvements for this task.

Third, we show that a novel distributed representation of syntactic dependencies between word pairs can be used to mine analogous dependencies from a large Chinese corpus. Inspired by the fact that distributed word representations can be used to measure word similarities, we use our distributed dependency representations to measure relation similarities. We propose a bootstrapping algorithm for analogy mining using dependency embeddings, and experiments on a large Chinese corpus show that the method can achieve a precision of 95.2% at a recall of 56.8%.

Our automatically-parsed corpus, trained embeddings and evaluation datasets are released publicly at `http://people.sutd.edu.sg/~yue_zhang/publication.html`. To our knowledge, we are the first to present results on Chinese analogy detection and to release large-scale Chinese word embeddings.

2 Background

2.1 Relational Similarity Tasks

There are three main tasks for relational similarity. This first is *relation classification*, which has been used in Task 2 of SemEval 2012 (Jurgens et al., 2012). In this task, all four words in two word pairs are given, and one needs to judge whether they belong to a same relation type. In order to address this task, various supervised methods have been used (Bollegala et al., 2008; Herdağdelen and Baroni, 2009; Turney, 2013).

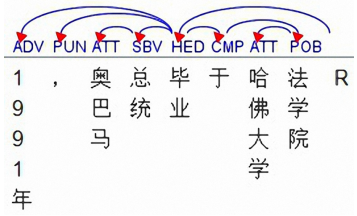

Figure 1: Dependency tree of the sentence "1991年 (in 1991) ， (,) 奥巴马 (Obama) 总统 (President) 毕业 (graduate) 于 (from) 哈佛 (Harvard) 法学院 (Law School)".

The second task is *analogy detection* (Mikolov et al., 2013b), which takes three words in two word pairs, and searches for a most suitable word from the vocabulary to recover the hidden word. This task has been addressed using word embeddings (Mikolov et al., 2013b; Levy and Goldberg, 2014b).

The third task is *analogy mining* (Chiu et al., 2007), which takes one word pair belonging to a certain semantic relation as a seed, and searches for all the word pairs that share the same relation with the seed. Compared with relation classification and analogy detection, analogy mining can be practically more useful because it requires less given information, and provides a large quantity of analogous word pairs automatically.

2.2 Skip-gram Word Embeddings

As a by-product of neural language models (Bengio et al., 2003; Mnih and Hinton, 2007), word embeddings are distributed vector representations of words, trained using local contexts. They capture linguistic regularities in languages (Mikolov et al., 2013b) and have been used in various tasks (Collobert and Weston, 2008; Turian et al., 2010; Socher et al., 2011).

In this paper, we apply the Skip-gram method of Mikolov et al. (2013a) for training embeddings, which works by maximizing the probability of a word given a context of multiple words. Mikolov et al. (2013b) use an ngram window as the context, and observe that the resulting embeddings are highly useful for unsupervised analogy detection.

2.3 Embedding-based Analogy Detection

Formally, the task of analogy detection is to find a word $b*$ given a pair of words *a:b* and a word $a*$ such that *a*:b** is analogous to *a:b*. Mikolov et al. (2013b) show that the task can be solved by finding a word that maximizes:

$$score = sim(b*, b - a + a*) \quad (1)$$

where sim is a similarity measure, typically the $cosine$ function. Levy and Goldberg (2014b) show that the Equation 1 is equivalent to:

$$score = cos(b*, b) - cos(b*, a) + cos(b*, a*) \quad (2)$$

As a result, the goal of analogy detection is to find a word $b*$ which is similar to b and $a*$ but different from a. Levy and Goldberg (2014b) further propose to substitute the addictive functions in Equation 2 with multiplicative functions:

$$score = cos(b*, b)cos(b*, a*)/(cos(b*, a) + \varepsilon) \quad (3)$$

Here $\varepsilon = 0.001$ is used to prevent division by zero. Their experiments show that the use of Equation 3 can improve the state-of-the-art. Following Levy and Goldberg (2014b), we refer to Equation 1 and 2 as 3COSADD and Equation 3 as 3COSMUL, respectively.

2.4 Chinese Relational Similarity

There are various types of relational similarities. Syntactically, inflections can be treated as a type of word-word relation (Mikolov et al., 2013b). For example, the comparative pairs "good:better" and "rough:rougher" are analogous, and the past tense inflections "see:saw" and "return:returned" are analogous. However, such inflectional relations do not apply to Chinese, which is fully configurational and lacks morphology. Consequently, our main focus is semantic similarities, which include antonymy (e.g. (热 (hot):冷 (cold)) VS (快 (fast):慢 (slow))), meronymy (e.g. (车 (car):轮子 (wheel)) VS (熊 (bear):掌 (paw))), gender (e.g. (男人 (man):女人 (woman)) VS (国王 (king):女王 (queen))) and function relations (e.g. (衣服 (clothing):穿 (wear)) VS (帽子 (hat):戴 (wear))), etc.

Chiu et al. (2007) show that English semantic relations are also reflected by syntactic dependencies. Their finding coincides with Levin (1993), who study English verbs. We find that this observation is even more prevalent for Chinese. In our automatically-parsed Chinese corpus of 3.4 billion words (Section 5.1), 86.4% word pairs from the analogy test dataset (Section 5.2) have corresponding dependencies, each of which appearing at least ten times.

The frequent correlation between semantic relations and syntactic dependencies can be due to the lack of morphology and function words in Chinese. In fact, Chinese syntactic ambiguities often need to be resolved by leveraging semantic information (Xiong et al., 2005; Zhang et al., 2014). Although not all occurrences of semantically-related word pairs must also form a syntactic dependency in a corpus, we show that syntactic dependencies can effectively improve analogy detection.

3 Dependency-context Word Embeddings for Analogy Detection

A first use of syntactic dependencies for embedding-based analogy detection is to use them directly for embeddings. Recently, a dependency context has been used for the skip-gram method, for capturing more syntactic regularities. Taking the sentence in Figure 1 for example, a *bi-gram* context for the word "毕业 (graduate)" can be "奥巴马 (Obama), 总统 (President), 于 (from), 哈佛 (Harvard)", while a *dependency context* of the same word can be "1991年/ADV, 总统/SBV, 于/CMP, 法学院/POB_于"[1], where "ADV, SBV, CMP, POB" indicate adverbial modifier, subject, complement and prepositional object, respectively.

It has been shown that a dependency context leads to embeddings that better help parsing (Bansal et al., 2014) and measuring word similarity (Levy and Goldberg, 2014a), compared with ngram contexts. However, little previous work has systematically compared dependency contexts with ngram contexts in analogy detection. We empirically study this problem (c.f Section 6.3), finding that dependency context leads to significantly worse analogy detection results for both Chinese and English using state-of-the-art embedding-based methods (Levy and Goldberg, 2014b). We give analysis in Section 6.4.

[1]The last token is a grand-child of "毕业 (graduate)", via the preposition "于 (at)" (Levy and Goldberg, 2014a).

4 Search Space Pruning Using Syntactic Dependencies

We study an alternative way of making use of syntactic dependencies, by using them to prune the vocabulary-sized search space of analogy detection. Given two word pairs *a:b* and *a*:b**, where *b** is hidden and *a* is the head word, we search for dependencies, taking *a** as the head word. The dependent words in the search candidates need to share the POS tag of *b*. If there are several types of dependencies between *a* and *b*, only the one with highest frequency is used. We rank all resulting dependencies using the 3COSMUL objective, and take the word *b** in the highest-scored dependencies as the answer.

For example, given the word pair (萨拉热窝 *(Sarajevo):*波黑 *(Bosnia and Herzegovina)*), whose most frequency dependency is <萨拉热窝 (Sarajevo), 波黑 (Bosnia and Herzegovina), ATT>, and the unknown pair (伦敦 *(London):b**), we acquire a list of dependencies, including <伦敦 (London), 美国 (USA), ATT>, <伦敦 (London), 巴黎 (Paris), COO>, <伦敦 (London), 加拿大 (Canada), ATT> and <伦敦 (London), 英国 (England), ATT>. Some of these dependencies, such as <伦敦 (London), 巴黎 (Paris), COO>, are parsed as the coordinate relation (COO), and thus pruned because the target syntactic relation is *ATT*. From the resulting list, the 3COSMUL objective successfully ranks the triple <伦敦 (London), 英国 (England), ATT> as the top candidate. In contrast, Levy and Goldberg's method takes "南非 (South Africa)" as the answer, which does not form an attributive-head phrase with "伦敦 (London)".

5 Analogy Mining Using Dependency Embeddings

Formally, *analogy mining* is the task of mining analogous dependencies $<x_1, y_1, r>, <x_2, y_2, r>$ ...$<x_n, y_n, r>$ that share the same relation *r* with a given dependency $<a, b, r>$. We mine analogous dependencies by considering relational similarity and attributional similarity simultaneously using the skip-gram model for embeddings.

5.1 Dependency Embedding

Inspired by the fact that *word* similarities can be measured by using distributed *word* representations, we hypothesize that *relation* similarities can be measured by distributed *relation* representations. Based on the observation in Section 2.4, semantically analogous word pairs typically have syntactic dependencies. We use the skip-gram algorithm to train *distributed representations of syntactic dependencies*, and use them for mining analogous word pairs.

With respect to the skip-gram model, *words* are the most common target for embeddings (Levy and Goldberg, 2014b; Levy and Goldberg, 2014a; Mikolov et al., 2013a), although continuous vector representations can be trained for other structures. For example, Mikolov et al. (2013a) take idiomatic *phrases* as embedding targets. *Dependencies*, which consist of a modifier word, a head word and a syntactic relation between them, can also be represented by continuous embeddings using the same algorithm.

To induce dependency embeddings, we take the union of the dependency context of both the dependent and the head of a dependency as the context. For instance, in the example sentence, the context of the dependency <总统 *(Presiden-*

```
Input  : dependency embedding DT, word
         embedding DW, seed dependency s,
         threshold α and β.
Output: set of ranked dependencies WP.
Function Mine (DT,DW,s,WP,α,β):
begin
    DTSet =∅;
    MScore =0;
    SimDT =GetSimDT (DT,s);
    for each Triple ∈ SimDT do
        MWS =GetMWord (s);
        HWS =GetHWord (s);
        MWD =GetMWord (Triple);
        HWD =GetHWord (Triple);
        ScoreX =Sim (MWS,MWD,DW);
        ScoreY =Sim (HWS,HWD,DW);
        ScoreXY =ScoreX × ScoreY;
        MScore =Max (ScoreXY,MScore);
        TopK (ScoreXY,Triple,DTSet,α)
    end
    MScore =MScore × β ;
    for each Triple, ScoreXY ∈ DTSet do
        if ScoreXY > MScore and Triple ∉
        WP then
            AddToSet (Triple,WP);
            s =Triple;
            Mine (DT,DW,s,WP,α,β);
        end
    end
end
WP =∅;
Mine (DT,DW,s,WP,α,β);
```

Algorithm 1: Bootstrapping for analogy mining.

t), 毕业 (graduate), SBV> consists of four tokens: "1991年/ADV", "奥巴马/ATT", "于/CMP" and "法学院/POB_于". The same skip-gram algorithm is used to train embeddings for dependency structures.

5.2 Analogy Mining by Bootstrapping

A bootstrapping algorithm is used to mine analogous word pairs based on dependency-context word embeddings and dependency embeddings. Algorithm 1 shows pseudocode of the recursive bootstrapping algorithm.

The recursive function `Mine` (Algorithm 1) contains three steps with six parameters, including the dependency embeddings *DT*, word embeddings *DW*, a seed dependency *s*, and two thresholds α and β. Step 1 (lines 3 to 5) is an initialization process, where the dependency embedding is used to return up to 100 most similar dependencies for the given seed *s*. These dependencies are stored in *SimDT*, and the candidate analogous dependency set *DTSet* is initialized to an empty set.

In Step 2 (lines 6 to 16), an analogous score *ScoreXY* is computed for each dependency *Triple* in *SimDT* by multiplying the similarity scores between the two dependents and the two heads in *Triple* and *s*, respectively. *Triple* is stored into the set *DTSet* if *ScoreXY* is ranked top α. The top 1 score in *DTSet* is referred to as *MScore*. In Step 3 (lines 17 to 24), if the score of a dependency *Triple* in *DTSet* is larger than $\beta \times$*MScore*, it is used as a new seed for mining more analogous dependencies, by calling the function `Mine` recursively.

We take the seed dependency <弹 (play), 钢琴 (piano), VOB> as an example to illustrate the work-flow of the `Mine` function. In Step 1, a set of similar dependencies (e.g., <弹 (play), 吉他 (guitar), VOB>, <弹 (play), 琴 (lyra), VOB>), is calculated using the dependency embeddings *DT* and stored in *SimDT*. Each dependency in *SimDT* is scored in Step 2, and the top α scores are put into the set *DTSet*. Finally, a dependency is used as seed to mine new analogous dependencies if its score is larger than a threshold ($\beta \times$*MScore*). For instance, the dependency <弹 (play), 琴 (lyra), VOB> is used to mine the new dependency <弹 (play), 古筝 (zheng), VOB>, which is then used to mine other dependencies such as <吹 (blow), 葫芦丝 (cucurbit flute), VOB> and <吹 (blow), 萨克斯 (sax), VOB>.

6 Experiments

6.1 Word Embeddings

We train three sets of word embeddings: NG5 (n-gram context with 5 words to the left of the target word and 5 words to the right), NG2 (2 words to the left and right) and DEP (dependency context), and one set of dependency embeddings DT (dependency context), using the Skip-Gram model. WORD2VEC[2] is used to train NG5 and NG2, and WORD2VECF[3] is used to train DEP and DT. The negative-sampling parameter is set to 15 in all the training processes.

All embeddings are trained on a free Chinese news archive[4] that contains about 170 millions sentences and 3.4 billions words. We segment and parse these sentences using the MVT implementation of ZPar 0.7[5] (Zhang and Clark, 2011), which is trained on a large-scale annotated corpus and achieves state-of-the-art analyzing accuracy on contemporary Chinese (Qiu et al., 2014)[6]. Targets and contexts for word and dependency embeddings were filtered with a minimum frequency of 100 and 10, respectively, and all the four types of embeddings are trained with 200 dimensions.

6.2 Datasets and Evaluation Metrics

Three datasets are used for evaluating Chinese embeddings. First, we construct a set of semantic analogy questions. This set contains five types of semantic analogy questions, including capital-country (136 word pairs, and 18354 analogy questions), provincial capital-province (28, 756), city-province (637, 386262), family member (male-female) (18, 306) and currency-country (62, 3782). We collect the five types of word pairs and then produce analogy questions automatically by concatenating two word pairs. The resulting analogy dataset contains 400K analogy questions. We refer to this dataset as the Chinese Analogy Question Set (CAQS).

[2] `http://code.google.com/p/word2vec/`

[3] `https://bitbucket.org/yoavgo/word2vecf`

[4] This dataset contains news articles in 2014 from various news websites, and can be downloaded from `http://pan.baidu.com/s/1o6wRjp4`

[5] `http://people.sutd.edu.sg/\%7Eyue_zhang/doc/doc/multiview.html`

[6] The system achieves 96.1% , 92.6% and 83.28% F1-score for words segmentation, joint POS-tagging and dependency parsing, respectively, on 1493 manually annotated sentences.

Data	Metrics	NG5	NG2	DEP
Cilin	P@1	43.3%	**45.9%**	43.6%
	P@5	31.1%	**33.3%**	32.6%
	P@10	25.5%	**27.5%**	**27.5%**
	P@20	20.5%	22.2%	**22.7%**
	P@50	15.0%	16.2%	**17.0%**
	P@100	11.5%	12.2%	**12.8%**
CWS	Kendall's τ	38.6%	**44.1%**	42.4%
	Spearman's ρ	54.5%	**62.2%**	60.7%

Table 1: Results on *Cilin* and CWS.

Because embeddings are central for analogy detection, yet there is little large-scale evaluation results on Chinese embeddings in the literature, we perform embedding evaluation on two datasets. The first one is the Chinese WordSim (CWS), translated from the English WordSim-353 Set and re-scored by native Chinese speakers (Jin and Wu, 2012). This dataset consists of 297 word pairs.

The second one is the Chinese thesaurus *Tongyicicilin* (Cilin) (Che et al., 2010), which groups 74,000 Chinese words into five-layer hierarchies and has been used for evaluating the accuracy of word similarity by traditional sparse vector space models (Qiu et al., 2011; Jin et al., 2012). The third level of *Cilin*, which contains 1428 classes, is used to evaluate whether two words are semantically similar.

For comparison between Chinese and English, we also use an English analogy question dataset, the Google dataset[7] (Mikolov et al., 2013a), to evaluate the English word embeddings of Levy and Goldberg (2014a)[8] on analogy detection.

On both the CAQS and the Google datasets, the 3COSMUL method (Levy and Goldberg, 2014b) is used to to answer analogy questions based on given embeddings. The results on the CWS dataset are evaluated using the two standard metrics for the task, namely Spearman's ρ and Kendall's τ rank correlation coefficients. The results on *Cilin* are evaluated using Precision@*K*: the percentage of words from the top-*K* candidates that belong to the *Cilin* category of the target word. If one of the top-*K* candidates belongs to the same third-level category in *Cilin* as the target word, the candidate word is taken as correct.

6.3 Dependency-based and Word-based Word Similarity and Analogy Detection

Word Similarity

[7] http://code.google.com/p/word2vec/source/browse/trunk/questions-words.txt

[8] http://levyomer.wordpress.com/2014/04/25/dependency-based-word-embeddings/

	Relation	NG5	NG2	DEP
MUL	capital-country	68.8%	52.7%	9.9%
	capital-province	84.0%	87.7%	50.0%
	city-province	80.9%	80.3%	22.6%
	family	39.7%	45.1%	41.5%
	currency	10.4%	9.9%	2.5%
	All	80.0%	78.8%	22.0%
IMP	capital-country	87.9%	88.0%	87.6%
	capital-province	84.9%	86.8%	84.9%
	city-province	91.8%	92.0%	90.5%
	family	45.3%	48.0%	47.1%
	currency	7.9%	7.0%	25.9%
	All	90.9%	91.1%	89.8%

Table 2: Results on CAQS. MUL and IMP indicate 3COSMUL and our improved method, respectively.

Relation	NG5	NG2	DEP
capital-country	94.6%	84.5%	38.5%
capital-world	71.5%	64.7%	14.2%
city-in-state	53.2%	42.5%	13.1%
family	82.0%	81.2%	81.0%
currency	10.5%	10.7%	6.0%
All	63.7%	60.7%	38.8%

Table 3: English results on the Google set.

Table 1 shows the results of the three Chinese embedding on *Cilin* and CWS, where NG2 performs much better than NG5 on both datasets. This demonstrates that one does not need to use large window sizes in training word-based embeddings for capturing word similarities. The result is similar to the finding of Shi et al. (2010), which indicates that a window size of 2 is better than a window size of 4 for capturing word similarity by using distributional word representations.

DEP performs slightly worse than NG2 on CWS and *Cilin* in P@1 and P@5. However, it achieves better results on *Cilin* in P@10 to P@100 when more candidate similar words are evaluated. In contrast, NG5 and NG2 mix more semantically related words. This finding is consistent with that of Levy and Goldberg (2014a).

Analogy Detection

Table 2 shows the results of the three Chinese embeddings on CAQS. Unlike on *Cilin* and CWS, NG5 outperforms DEP, and is also slightly better than NG2. Similar tendency is shown in Table 3 for the three English embeddings evaluated on the Google dataset. These results show that dependency embeddings are relatively weak for answering analogy questions. On the other hand, the performance also varies across different relation types.

Target	NG5	NG2	DEP
穿 (wear)	短裤 (shorts), 紧身 (slim-fit), 身穿 (wear), 外套 (coat), 裙子 (skirt)	身穿 (wear), 身着 (wear), 短裤 (shorts), 戴 (wear), 紧身 (slim-fit)	身穿 (wear), 身着 (wear), 戴 (wear), 改穿 (change cloths), 外穿 (wear outside)
关羽 (Guan Yu; P)	赵云(Zhao Yun; P), 刘备 (Liu Bei; P), 诸葛亮 (Zhuge Liang; P), 张飞 (Zhang Fei; P), 曹操 (Cao Cao; P)	赵云 (Zhao Yun; P), 刘备 (Liu Bei; P), 张飞 (Zhang Fei; P), 曹操 (Cao Cao; P), 夏侯渊 (Xiahou Yuan; P)	赵云 (Zhao Yun; P), 韩信 (Han Xin; P), 曹操 (Cao Cao; P), 刘备 (Liu Bei; P), 阿修罗 (Asura; P)
郑州 (Zhengzhou; C)	石家庄 (Shijiazhuang; C), 洛阳 (Luoyang; C), 西安 (Xian; C), 许昌 (Xuchang; C), 太原 (Taiyuan)	石家庄 (Shijiazhuang; C), 太原 (Taiyuan; C), 济南 (Ji-nan; C), 合肥 (Hefei; C), 西安 (Xi-an; C)	合肥 (Hefei; C), 济南 (Jinan; C), 武汉 (Wuhan; C), 石家庄 (Shijiazhuang; C), 南宁 (Nanning; C)

Table 4: Comparison between NG2, NG5 and DEP Embeddings. (P: personal name, C: city name)

6.4 Analysis

To analyze the difference between the three Chinese embeddings methods qualitatively, we manually inspect the words "穿 (wear)", "关羽 (Guan Yu, a person name in the novel '三国演义 *(Romance of the three kingdoms)*')", and "郑州 (Zhengzhou, a city)". Their most similar words are shown in Table 4.

Word Similarity

For the word "穿 (wear)", both NG5 and NG2 yield *similar words* such as "身穿 (wear)", "身着 (wear)", "戴 (wear)" and *related words* such as "短裤 (shorts)", "紧身 (slim-fit)", "外套 (coat)", "裙子 (skirt)", although NG5 gives more related words. In contrast, DEP gives only words that are *similar* both syntactically and semantically. This observation holds for other verbs and nouns, and can be explained by the context extraction methods. For instance, the word "穿 (wear)" usually takes one of the words "短裤 (shorts)", "外套 (coat)", "裙子 (skirt)" as its object, and thus shares similar contexts with them in NG5 and NG2. The context extraction method in DEP, on the other hand, yields different context across syntactic roles, such as verbs (e.g. "穿 (wear)") and their objects (e.g. "短裤 (shorts)" and "外套 (coat)").

Observations on the person name "关羽 (Guan Yu)" and location "郑州 (Zhengzhou)" are similar. For "关羽 (Guan Yu)", NG5 and NG2 can yield more person names in the same novel, while DEP yields person names from other novels (i.e. "韩信 (Hanxin)" and "阿修罗 (Asura)"). For "郑州 (Zhengzhou)", the provincial capital of "河南 (Henan)", NG5 and NG2 give more cities in the same province "河南 (Henan)", while DEP yields capitals of other provinces.

Analogy Detection

As mentioned in Section 2.3, both 3COSADD and 3COSMUL seek a word $b*$ that is similar to b and $a*$ but dissimilar to a. Ideally, the two word pairs b:$b*$ and a:$a*$ should be semantically *similar* while the two word pairs a:b and $a*$:$b*$ should be semantically *related*. Therefore, 3COSADD and 3COSMUL require the embeddings to give higher cosine scores for both semantically similar and related words.

Our analysis above shows that word-context embeddings tend to mix semantically *related* and *similar* words, but dependency-context embeddings only capture semantic *similarity*. This partly explains the reason that dependency-context word embeddings are weak for analogy detection.

It has also been shown in Section 6.3 that the performances of analogy detection vary across different types of relations, which indicates that there are more sophisticated underlying factors. One intuitive explanation is that different semantic relations correspond to different syntactic dependency structures. For example, the male-female family member relation is expected to stand less frequently in a syntactic dependency relation, compared with geographic relations such as city-country, which stand frequently in attributional syntactic relations (e.g. "London, England"). As a result, where the coupling between syntactic and semantic relations is weak, our analysis in Section 6.3 and other work based on syntactic relations can find limitations.

6.5 Syntactic Dependencies for Improved Analogy Detection

The results on CAQS using the method in Section 4 are shown in the IMP rows of Table 2. The method achieves significant improvements (from 80.0% to 90.9% using NG5) compared with Levy and Goldberg's method. In addition, DEP also performs significantly better than with MUL, with an increase from 22.0% to 89.8%. The main reason for this improvement is that the filtering process using syntactic dependencies successfully prunes noisy words.

Seed	Count	Prec
吃 (eat), 苹果 (apple), VOB	572	84.70%
弹 (play), 钢琴 (piano), VOB	142	40.49%
穿 (wear), 衣服 (clothing), VOB	452	67.37%
写 (write), 小说 (novel), VOB	441	53.40%
中国 (China), 北京 (Beijing), ATT	2224	95.23%
湖北 (Hubei), 武汉 (Wuhan), ATT	3201	96.34%

Table 5: Main results of Analogy Mining.

Error analysis shows that the main errors by the improved method are quite different from those by the baseline. For instance, the main errors of Levy and Goldberg's method for the city-province relation are caused by giving another province as the answer, while the improved method gives the name of the country as answer. This is because irrelevant provinces do *not* co-occur frequently with the city in syntactic dependencies, and hence can be filtered by our method. On the other hand, both the country name and province name co-occur frequently with the city name in syntactic dependencies, and our method cannot make a choice between them.

6.6 Dependency Structure Embeddings for Analogy Mining

Shown in Table 5, we use six seeds to mine analogous dependencies. The first seed is used for development and the others for test. The first three seeds, the fourth seed and the last two seeds belong to the *Use:Thing, Produce:Thing and Sub-Location:Location* relations, respectively. α and β are set to 20, and 0.6, respectively. Each set of mined dependencies together with the seed dependency and relation type is shown to two human evaluators, who are required to give a Yes/No answer to each dependency in the set. We take the average scores of the two evaluators (the average inter-annotator agreement is 0.95) as the final precision scores.

As shown in the table, the precisions using different seeds are quite different, ranging from 40% to 96%. One possible reason is that different relations have different numbers of analogous dependencies, ranging from dozens to thousands, and thus the fixed thresholds tuned on a development seed does not apply as effectively to all test cases. For instance, "弹 (play)" and its analogous actions, "吹 (blow)" and "拉 (play)", are all human actions on musical instruments, while the actions "吃 (eat)" and "写 (write)" can apply to many patients. For the seed <弹 (play), 钢琴 (piano), VOB>, irrelevant results such as <拿 (use), 剪子 (scissors), VOB> and <拿 (use), 电筒 (flashlight), VOB>, have the verb "拿 (use)", which is also a human action, yet cannot be considered as usage of the patients "剪子 (scissors)" and "电筒 (flashlight)". Because of the stricter selectional preference of "弹 (play)", its precision of analogy mining is lower.

We tentatively measure the recall of the algorithm by taking the first three types of word pairs in CAQS as the gold set, which contains 801 word pairs. All the three types of word pairs belong to the relation *Sub-Location:Location*. The recall is computed as the percentage of the gold word pairs covered by the mined dependencies. When using the two seeds <武汉 (Wuhan), 湖北 (Hubei), ATT> and <北京 (Beijing), 中国 (China), ATT> for analogy mining, the recalls are 50.2% and 11.3%, respectively. Their union recall is 56.8%. When the precision of each seed is similar, we can achieve better recall without precision loss by using more seeds.

7 Related Work

Turney (2006) introduces a latent relational analysis (LRA) model to measure relational similarity, and apply a novel co-occurrence-based method for analogy filtering. The model can be used for both analogy detection and relation classification, yet cannot scale up well to large datasets due to the complexity of Singular Value Decomposition. Recently, distributed word representations using the skip-gram model (Mikolov et al., 2013a) has been shown to give competitive results on analogy detection. Levy and Goldberg (2014a) extends the skip-gram method with dependency-context embeddings. We study the effect of Levy and Goldberg's embeddings on analogy detection, and further extend their embeddings to dependency-context *dependency structure* embeddings for analogy mining.

Chiu et al. (2007) presents a similarity graph tranversal (SGT) method to mine analogous relations from raw English text automatically, using syntactic dependencies to find candidate relations. The method is unsupervised, and can scale up well to large data sets. However, Chiu et al. (2007) mainly focuses on relations between subjects and objects because of its word-pair extraction method. Ó Séaghdha and Copestake (2009) is a supervised method, which combines lexical similarity and relational similarity to classify se-

mantic relations. These methods are based on distributional word representation models and fit for classifying noun-noun word pairs. In contrast, our methods are based on distributed word representation models, and can mine noun-noun word pairs as well as verb-noun word pairs. In addition, our analogy mining method is unsupervised, while the methods of both Turney (2006) and Ó Séaghdha and Copestake (2009) are supervised.

8 Conclusion

We studied several Chinese relational similarity tasks to train embeddings under the context of distributed word representations using the skip-gram model and syntactic dependencies. For Chinese analogy detection, we compared word-context and dependency-context embeddings, finding that the former results in much better accuracies. Observing that common relations in Chinese are frequently represented by syntactic dependencies, we improved Chinese analogy detection using a dependency context. Further, we empirically studied Chinese analogy mining by proposing a bootstrapping algorithm using a novel distributed representation of syntactic dependencies.

Acknowledgments

We thank the anonymous reviewers for their constructive comments, and gratefully acknowledge the support of the Singapore Ministry of Education (MOE) AcRF Tier 2 grant T2MOE201301, the National Natural Science Foundation of China (No. 61572245, 61170144, 61103089), Major National Social Science Fund of China (No. 12&ZD227), Scientific Research Foundation of Shandong Province Outstanding Young Scientist Award (No. BS2013DX020) and Humanities and Social Science Projects of Ludong University (No. WY2013003).

References

Mohit Bansal, Kevin Gimpel, and Karen Livescu. 2014. Tailoring continuous word representations for dependency parsing. In *Proceedings of ACL*, Baltimore, Maryland, June.

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. 2003. A neural probabilistic language model. *Journal of Machine Learning Research*, 3:1137–1155.

Danushka Bollegala, Yutaka Matsuo, and Mitsuru Ishizuka. 2008. Www sits the sat: Measuring relational similarity on the web. In *ECAI*, pages 333–337.

Wanxiang Che, Zhenghua Li, and Ting Liu. 2010. Ltp: A chinese language technology platform. In *Proceedings of COLING*, pages 13–16.

Andy Chiu, Pascal Poupart, and Chrysanne DiMarco. 2007. Generating lexical analogies using dependency relations. In *Proceedings of EMNLP-CoNLL 2007*, pages 561–570, Prague, Czech Republic, June.

Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of ICML*, pages 160–167. ACM.

Amaç Herdağdelen and Marco Baroni. 2009. Bagpack: A general framework to represent semantic relations. In *Proceedings of the Workshop on Geometrical Models of Natural Language Semantics*, pages 33–40.

Peng Jin and Yunfang Wu. 2012. Semeval-2012 task 4: evaluating chinese word similarity. In *Proceedings of the Sixth International Workshop on Semantic Evaluation*, pages 374–377.

Peng Jin, John Carroll, Yunfang Wu, and Diana McCarthy. 2012. Distributional similarity for chinese: Exploiting characters and radicals. *Mathematical Problems in Engineering*, 2012.

David A Jurgens, Peter D Turney, Saif M Mohammad, and Keith J Holyoak. 2012. Semeval-2012 task 2: Measuring degrees of relational similarity. In *Proceedings of the Sixth International Workshop on Semantic Evaluation*, pages 356–364. Association for Computational Linguistics.

Beth Levin. 1993. English verb classes and alternations: a preliminary investigation.

Omer Levy and Yoav Goldberg. 2014a. Dependency-based word embeddings. In *Proceedings of ACL*, pages 302–308, Baltimore, Maryland, June.

Omer Levy and Yoav Goldberg. 2014b. Linguistic regularities in sparse and explicit word representations. In *Proceedings of CONLL*, pages 171–180, Ann Arbor, Michigan, June.

Douglas L Medin, Robert L Goldstone, and Dedre Gentner. 1990. Similarity involving attributes and relations: Judgments of similarity and difference are not inverses. *Psychological Science*, 1(1):64–69.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. 2013b. Linguistic regularities in continuous space word representations. In *HLT-NAACL*, pages 746–751. Citeseer.

Andriy Mnih and Geoffrey Hinton. 2007. Three new graphical models for statistical language modelling. In *Proceedings of ICML*, pages 641–648. ACM.

Diarmuid Ó Séaghdha and Ann Copestake. 2009. Using lexical and relational similarity to classify semantic relations. In *Proceedings of EACL 2009*, pages 621–629, Athens, Greece, March.

Likun Qiu, Yunfang Wu, and Yanqiu Shao. 2011. Combining contextual and structural information for supersense tagging of chinese unknown words. In *Computational Linguistics and Intelligent Text Processing*, pages 15–28. Springer.

Likun Qiu, Yue Zhang, Peng Jin, and Houfeng Wang. 2014. Multi-view chinese treebanking. In *Proceedings of COLING*, pages 257–268, Dublin, Ireland, August.

Shuming Shi, Huibin Zhang, Xiaojie Yuan, and Ji-Rong Wen. 2010. Corpus-based semantic class mining: Distributional vs. pattern-based approaches. In *Proceedings of COLING*, pages 993–1001, Beijing, China, August.

Richard Socher, Cliff C Lin, Chris Manning, and Andrew Y Ng. 2011. Parsing natural scenes and natural language with recursive neural networks. In *Proceedings of ICML*, pages 129–136.

Joseph Turian, Lev Ratinov, and Yoshua Bengio. 2010. Word representations: a simple and general method for semi-supervised learning. In *Proceedings of ACL*, pages 384–394.

Peter D Turney and Michael L Littman. 2005. Corpus-based learning of analogies and semantic relations. *Machine Learning*, 60(1-3):251–278.

Peter D Turney. 2006. Similarity of semantic relations. *Computational Linguistics*, 32(3):379–416.

Peter D Turney. 2013. Distributional semantics beyond words: Supervised learning of analogy and paraphrase. *arXiv preprint arXiv:1310.5042*.

Deyi Xiong, Shuanglong Li, Qun Liu, Shouxun Lin, and Yueliang Qian. 2005. Parsing the penn chinese treebank with semantic knowledge. In *Natural Language Processing–IJCNLP 2005*, pages 70–81. Springer.

Yue Zhang and Stephen Clark. 2011. Syntactic processing using the generalized perceptron and beam search. *Computational Linguistics*, 37(1):105–151.

Meishan Zhang, Yue Zhang, Wanxiang Che, and Ting Liu. 2014. A semantics oriented grammar for chinese treebanking. In *Computational Linguistics and Intelligent Text Processing*, pages 366–378. Springer.