Real Multi-Sense or Pseudo Multi-Sense:
An Approach to Improve Word Representation

Haoyue Shi[1], Caihua Li[1] and Junfeng Hu[1,2*]
[1] School of Electronics Engineering and Computer Science,
Peking University, Beijing, China
[2] Key Laboratory of Computational Linguistics,
Ministry of Education, Peking University, Beijing, China
{hyshi,peterli,hujf}@pku.edu.cn

Abstract

Previous researches have shown that learning multiple representations for polysemous words can improve the performance of word embeddings on many tasks. However, this leads to another problem. Several vectors of a word may actually point to the same meaning, namely pseudo multi-sense. In this paper, we introduce the concept of pseudo multi-sense, and then propose an algorithm to detect such cases. With the consideration of the detected pseudo multi-sense cases, we try to refine the existing word embeddings to eliminate the influence of pseudo multi-sense. Moreover, we apply our algorithm on previous released multi-sense word embeddings and tested it on artificial word similarity tasks and the analogy task. The result of the experiments shows that diminishing pseudo multi-sense can improve the quality of word representations. Thus, our method is actually an efficient way to reduce linguistic complexity.

1 Introduction

Representing meanings of words by embedding them into a high dimensional vector space, so called word embedding, is a useful technique in natural language processing. An intuitive idea is to encode one word into a single vector, which contains the semantic information of the word in corpus (Bengio et al., 2003; Collobert and Weston, 2008; Mnih and Hinton, 2007; Mikolov et al., 2010).

There is a consensus that natural languages always include lots of polysemous words. For example, when the word *star* appears together with words like *planet, satellite*, it may roughly denote a kind of celestial body; when *star* appears with words like *movie, song, drama*, it may stand for a famous person. For most cases, we human beings can easily point out which sense a word belongs to based on its context. Considering the polysemous words, some previous approaches have learned multiple embeddings for a word, discriminating different senses by their context, related syntax and topics (Reisinger and Mooney, 2010; Huang et al., 2012; Chen et al., 2014; Pina and Johansson, 2014; Neelakantan et al., 2015; Cheng and Kartsaklis, 2015; Liu et al., 2015). The authors also provided methods to disambiguate among the multiple representations. Li and Jurafsky (2015) have demonstrated that multi-sense word embeddings could be helpful to improve the performance on many NLP and NLU tasks.

However, this leads to another problem. It's much more difficult for computer than human beings to detect whether two appearances of a same word stand for the same sense. Moreover, the contexts may be totally different even if these appearances belong to the same meaning based on human judgement. Previous multi-sense word embedding approaches often tend to embed a word in such situation into more than one vector by mistake (actually, they have the same meaning and should be embedded into only one vector). Consider three different representations of word *bear* learnt by the method introduced by Neelakantan et al. (2015), which are shown by their nearest neighbors in the vector space *MSSG-50d*.

- emerald, **bears**, **three-toed**, **snake**, **periwinkle**, **ruffed**, **hoopoe**, distinctive, unmistakable

- **bird**, **wolf**, arrow, **pelican**, emerald, canyon, diamond, **buck**, **deer**

- pride, lady, hide, king, gift, crane, afflict, promise, reap, protect

The words clearly related to the domain *animals* are bolded. We could infer that the first two representations have the same meaning that points to the animal bear, and the third representation has different meaning. We call such different learnt representations of a word with the same meaning (e.g. the first two representations of word *bear* shown above) *pseudo multi-sense*, where we judge whether senses are pseudo multi-sense by comparing their domains.

Given the word embeddings, which have multiple vectors for each polysemous word, we introduce an algorithm based on domains and semantic relations to detect pseudo multi-sense, since word representations which stand for the same meaning would have the same hypernym and belong to the same domain. Then we try to eliminate the effect of pseudo multi-sense by training a global transition matrix which projects the original word vectors into a new vector space based on the detected pseudo multi-sense pairs, minimizing the distance between pseudo multi-sense pairs in the vector space while keeping the spatial relation of other pairs. We propose the algorithm in Section 3 and evaluate it in Section 4.

Obviously, detecting and diminishing pseudo multi-sense would make word sense representations, which can be processed by computer, closer to human thinking. We also suggest this approach can improve the performance on real world NLU tasks by evaluating the algorithm on the analogy test dataset introduced by Mikolov et al. (2013a), and also on WordSim-353 (Finkelstein et al., 2001) and SCWS (Huang et al., 2012) dataset which include human judgements on similarity between pairs of words.

2 Background and related work

2.1 Distributional word representations

Since Bengio et al. (2003) applied neural network to language model, which treats word embeddings as parameters and thus it allows us to learn the language model and word embeddings at the same time, many researchers have proposed other neural network models (Mnih and Hinton, 2007; Collobert and Weston, 2008; Mikolov et al., 2013a) to improve in both efficiency and accuracy. What's more, hierarchical softmax by Morin and Bengio (2005), noise contrastive estimation by Mnih and Kavukcuoglu (2013) and negative sampling by Mikolov et al. (2013c) make it possible to learn accurate word embeddings in a short time.

2.2 Multi-sense word embeddings

Most vector-space models (VSMs) represent a word with only one vector, which clearly fails to capture homonymy and polysemy. And thus, Huang et al. (2012) proposed a method to generate the context embeddings in the following way. Firstly, they generate single-sense word embeddings and compute out the context embeddings. Then they cluster the context embeddings, and the result are used to re-label each occurrence of each word in the corpus. Thirdly, the model they proposed is applied to the labeled corpus to generate the multi-sense embeddings. Chen et al. (2014) took external knowledge base into consideration and built a model to learn a separate vector for each sense pre-defined by WordNet (Miller, 1995). Neelakantan et al. (2015) improved multi-sense word embedding model by dropping the assumption that each word should have the same number of senses, and proposed a non-parametric model to automatically discover a varying number of senses per word type. Cheng and Kartsaklis (2015) proposed a syntax-aware approach for multi-sense word embeddings.

2.3 WordNet and WordNet domain knowledge

WordNet (Miller, 1995) is a large lexical database of English. Nouns, verbs, adjectives and adverbs are grouped into sets of cognitive synonyms, namely synsets, each expressing a distinct concept. Synsets are represented by a word, a pos tag and a label, and interlinked by means of conceptual-semantic and lexical relations (hypernymy/hyponymy). Chen et al. (2014) used WordNet to improve word embeddings.

Magnini and Cavaglia (2000) and Bentivogli et al. (2004) presented a WordNet Domains Hierarchy, which is a language-independent resourse composed of 164 domain labels. What's more, González et al. (2012) provided a graph based improvement and released a domain knowledge (Extended WordNet

Domains) base aligned to WordNet 3.0, which we use in our experiments as domain knowledge. Extended WordNet Domains contains 170 domains and the probability of each synset in WordNet 3.0 in each domain. The domains it provided include *acoustics, agriculture, volleyball, etc.*

2.4 Vector space projection

Even though bilingual data always plays an important role in the modern statistical machine translation system, it had failed to map the missing word and phrase entries between two languages until Mikolov et al. (2013b) proposed a simple but effective method to extend dictionaries and translation tables. The main idea of this novel method is to learn a linear projection between the languages using a small bilingual dictionary but making little assumption about the languages, which has proved to be able to project the vector representation of any word from the source space to the target space accurately. Our vector space projection algorithm is very similar to this.

3 Pseudo multi-sense detection and elimination by vector space projection

3.1 Domain based pseudo multi-sense detection

3.1.1 Direct domain similarity

Given a word and its context, we human beings can easily determine the domains this word belongs to. WordNet makes it convenient for users to get the domains of all synsets of a word. To determine the domain of a sense given the multi-sense word embeddings, we can intuitively define the probability that the kth sense of word w belongs to domain d as

$$P_D(w, k, d) \propto \sum_{w' \in NN(w,k)} D(p(w'), d) \tag{1}$$

where $NN(w, k)$ is the nearest neighbors of the kth sense of word w in the given word embeddings, $p(w')$ is the protocol representation of word w' (e.g. when w' is *star_s1*, $p(w')$ would be *star*), $D(p(w'), d)$ is the sum probability that domain d appears in all synsets of $p(w')$ in WordNet provided by Extended WordNet Domain. Then we can compute the domain similarity between the kth and the lth sense of word w by

$$Sim_D(w, k, l) = \frac{1}{n} |TopN(P_D, w, k, n) \cap TopN(P_D, w, l, n)| \tag{2}$$

where $TopN(P, w, k, n)$ is the set of x that $P(w, k, x)$ ranks top n in decreasing order (in our experiments, $n = 5$).

3.1.2 Semantic hierarchical similarity

However, in the knowledge base we applied, the domain knowledge is sometimes not enough for dectecting pseudo multi-sense, especially for some abstract words. For example, it's hard to specify which domain the word *extract* belongs to. What's more, based on González et al. (2012), the Extended WordNet Domain cannot reach the precision of 100%. So we tend to apply semantic hierarchy, particularly hypernymy relations, to help improve our pseudo multi-sense detecting as supplement, since hypernymy somehow contains some domain information. With WordNet, we can also get the semantic relations (e.g. hypernymy, hyponymy, synonymy) of synsets. With the consideration of the DAG structure of semantic relations, for hypernyms of a specific word, the nearer the hypernym, the more information it contains. So we penalize the *far hypernyms*, like *whole, entity, thing*, which cover a large amount of words as their hyponyms. Similar to the definition of $P_D(w, k, t)$, we can define the probability that the kth sense of word w has the hypernym h, where h is a synset in WordNet, as

$$P_H(w, k, h) \propto \frac{1}{d(w, h)} \sum_{w' \in NN(w,k)} H(p(w'), h) \cdot \frac{1}{d(p(w'), h)} \tag{3}$$

where $d(w, h) = \min_{sw \in Synsets(w)} dis(sw, h)$, $dis(x, y)$ is the distance between two synsets x and y in WordNet, $H(p(w'), h)$ is the frequency that the synset h appears as a hypernym of a synset of $p(w')$ in WordNet. In particular, if h is not a hypernym of w in WordNet, $P_H(w, k, h) = 0$.

We then compute the semantic hierarchical similarity between the kth and the lth sense of word w by

$$Sim_H(w,k,l) = \frac{1}{n}|TopN(P_H, w, k, n) \cap TopN(P_H, w, l, n)| \qquad (4)$$

With the definition of domain similarity and semantic hierarchical similarity, we can compute the similarity between the kth and the lth sense of word w by

$$Sim(w,k,l) = Sim_D(w,k,l) + Sim_H(w,k,l) \qquad (5)$$

When $Sim(w,k,l) > \lambda$, where λ is a hyper-parameter ($\lambda = 1$ in our experiments), we consider the kth and the lth sense of word w have the same meaning. In other words, we are able to detect pseudo multi-sense pair (w_k, w_l) based on $Sim(w,k,l)$, which is called pseudo multi-sense detection.

3.2 Pseudo multi-sense elimination

Having the existing word embeddings, assume that we have a detected pseudo multi-sense group $G = \{w_{k_1}, w_{k_2}, ..., w_{k_n}\}$, in which $w_{k_1}, w_{k_2}, ..., w_{k_n}$ are senses of word w, taking the same meaning. Thus, we can find a representative vector for the group. Let $v_s(w, k_i)$ be the corresponding vectors of w_{k_i}, and $v_r(G)$ be the representative vector for the group G. Such vector $v_r(G)$ can be randomly chosen from $\{v_s(w, k_1), v_s(w, k_2), ..., v_s(w, k_n)\}$, or simply the mean vector of them. Other methods to compute $v_r(G)$ are also worth trying if reasonable.

Inspired by Mikolov et al. (2013b), we assume there is a transition matrix, by which for all pseudo multi-sense group G, $\forall w_{k_i} \in G$, $v_{w_{k_i}}$ can be projected to $v_r(G)$. The experiments shown in Section 4 supported our assumption. In other words, we suggest that there exists a global matrix Φ, for any given pseudo multi-sense group $G = \{w_{k_1}, w_{k_2}, ..., w_{k_n}\}$ and its representative vector $v_r(G)$, we have

$$v_r(G) = \Phi * v_s(w, k_i), \forall w_{k_i} \in G, \forall G \qquad (6)$$

Stochastic gradient descent (SGD) is a stochastic approximation of the gradient descent optimization method for minimizing an objective function written as a sum of differentiable functions by iteration. In order to obtain a consistent Φ for the projection of all pseudo multi-sense group, we can learn an approximate Φ with SGD for optimization. Then we use the obtained Φ to project existing word embeddings, and thus we can get a new vector space in which pseudo multi-sense has been eliminated compared to the original space.

4   Experiments

We evaluate our pseudo multi-sense detecting and eliminating method both qualitatively and quantitatively. We apply our method to the released word embeddings by Huang et al. (2012) and Neelakantan et al. (2015), which were both trained on the same Wikipedia corpus, and display the performance of our method based on the nearest neighbor task, word similarity tasks and the analogy task. In the following parts, MSSG and NP-MSSG are word embeddings released by Neelakantan et al. (2015); 50d and 300d are the dimensions of the vector space. The vector space released by Huang et al. (2012) are 50-dimensional.

4.1   Nearest Neighbors

As we hypothesized, previous multi-sense word embedding methods would produce a lot of pseudo multi-sense examples. For the convenience of view, we only focus on the semantic relation in the qualitative evaluation part. We extracted the most probable hypernym for each sense of some sample words by Eq(4), using the synset semantic relations provided by WordNet (Miller, 1995). If different representations of one word have the same hypernym, we consider them as pseudo multi-sense.

In Table 1, we show the nearest neighbors for each sense of each sample word with multiple word embeddings and our result of pseudo multi-sense detecting. For most of the representations, according to their nearest neighbors, we got reasonable hypernyms. However, there are also some unexpected cases

STAR

Huang et al.	princess, series, cast, serial, midway, sparkle, 1940s, leo, closet, co-star	01
	silver, boy, cat, version, adventures, stars, emerald, destroyer, terrace, planet	02
	energy, disk, wheel, disadvantage, block, puff, radius, diamond, chord	03
	version, bronze, standard, colors, ring, emblem, silver, wear, shoulder, red	01
	workshop, shop, paper, merchandise, plain, corporation, stock, likeness	03
	guard, baseball, starter, tennis, basketball, brazil, class, world, morocco, ncaa	01
	appearance, entertainer, pat, alumnus, freelance, brother, session, receiver	01
	fictional, ongoing, manga, super, japanese, silver, interactive, asian, fiction	01
	die, express, ride, opera, spanish, musical, hour, disaster, sun, blue	01
	galaxy, spiral, variable, guide, magnitude, companion, satellite, crater	02
MSSG-50d	blue, dragon, acbl, diamond, purple, legion, arrow, mercury, eagle, cross	01
	fan, legend, show, moesha, heroes, guest-star, flicka, lassie, tv-movie	01
	stars, sun, constellation, galaxy, eridani, pegasi, supergiant, ceti, starburst	02

01: person.n.01   02: celestial_body.n.01   03: whole.n.02

ROCK

Huang et al.	blur, indulgence, pop, noise, bands, lacuna, reformed, wave, genre, taster	01
	energy, silver, cat, song, cd, planet, dawn, hero, video, terrace	02
	metal, classic, legendary, dubbed, american, hard, belgian, short-lived, debut, da	01
	soft, shifting, disappear, fill, crystalline, false, pitch, expanse, heat, pile	03
	vinyl, concert, limited, box, summer, double, dance, enhanced, gold, inch	04
	hop, well-known, folk, occasional, jazz, music, concert, array, hard, pop	01
	morris, miami, wood, ghost, silver, pearl, chase, corner, oak, thousand	03
	hard, pop, cm, jazz, hip, hop, r&b, gutter, wave, subculture	01
	hard, hip, short-lived, classic, jazz, raw, metal, ep	01
	jazz, rally, star, roll, live, entertainer, appearance, session, pop, cover	01
MSSG-50d	metal, rippling, dense, swirling, chirping, blues, punk, psychedelia, bands, pop	01
	sand, rocks, butte, ash, sandy, little, cedar, rocky, sugarloaf, spring-fed	03
	hip, alternative, indie, progressive, hop, reggae, roll, rock/metal, post-hardcore	01

01: popular_music.n.01   02: person.n.01   03: material.n.01   04: whole.n.02

NET

Huang et al.	reduction, amount, increases, stamina, zero, worksheet, improvements, sum	01
	raw, atomic, destination, brave, orbit, generalize, clock, ca, exhale, fresh	02
	monthly, minimum, retail, banking, dividend, investor, tax, consumer, flat, dollar	03
	cash, annual, bribe, yen, generate, yen, liabilities, stocks, lifetime	03
	limousine, panic, alarm, cotton, racket, rush, 9th, buffalo, corps, recovered	04
	palm, stalk, blanket, challah, qibla, putting, recess, curtain, tighten, lean	04
	indent, text, poser, instruction, libraries, mosaic, campaigns, graphics, imperative	04
	freight, processing, volume, needs, passenger, junction, electrical, ferry, shipping	04
	contribution, bonus, compensation, bribe, yen, liabilities, stocks, yen, profit	03
	1909, quarterback, columbus, bills, bath, elite, 1903, tigers, affiliated, eagles	04
MSSG-50d	droplet, pile, wellbore, squeeze, amount, volume, steady, turn, moves, balance	04
	boards, run, ball, spot, sideline, at-bat, clock, stretch, running, phils	04
	revenue, trillion, assets, profit, billion, pre-tax, liabilities, index, us$, fdi	03

01: whole.n.02   02: seize.v.01   03: income.n.01   04: artifact.n.01

Table 1:   Nearest neighbors (by cosine similarity) of sample words and the result of pseudo multi-sense detecting. Column 1 shows the existing word embeddings we use to detect pseudo multi-sense. In Column 2, each row shows the nearest neighbors of one sense in the vector space (Column 1). In Column 3, we present a meaning label for each sense, following the standard of WordNet synset description. We argue that "senses" with the same label actually have the same meaning, namely pseudo multi-sense.

from the result based on the word vectors released by Huang et al. (2012), while no such cases are found in the vectors released by Neelakantan et al. (2015). For example, we got [whole.n.02] as the hypernym of the three sample words (which seems too general since *whole* can be the hypernym of nearly all entities), and [person.n.01] as a hypernym of *ROCK* (which seems not very reasonable according to the nearest neighbors). By intuition, we suggest that is because of the quality of the word embeddings. Possibly, the level of confidence to extract domains and hypernyms for each sense could be a metric for evaluating the quality of word embeddings. From this point of view, the word embeddings released by Neelakantan et al. (2015) are also with higher quality.

4.2 Word Similarity

Now we focus on applying a qualitative evaluation to our method. For each word in the embedded vector space, we first determine the pseudo multi-sense with Eq(5). Then we try to minimize the distance between vectors which belong to the same pseudo multi-sense group, since we argue that they actually represent for the same meaning in the vector space, by training such a matrix Φ, which projects all vectors to a new vector space and eliminate the distance between pseudo multi-sense vectors. We train the matrix Φ by minimizing the following formula.

$$L = \sum_{(x,x_r)} ||\Phi x - x_r||^2 \tag{7}$$

where x is a vector which belongs to a pseudo multi-sense group and x_r is the representative vector of the corresponding group. In our experiments, we tried both random sampling and computing mean vector for getting such representative vector.

4.2.1 Similarity Metrics

The similarity here is a metric between words to evaluate the performance of word embeddings, which will be used to compare with human judgements, differently from the similarities we introduced in Section 3, which are used to detect pseudo multi-sense.

Neelakantan et al. (2015) introduced three metrics to compute the similarity between words in multi-sense word embeddings, which are $avgSim, avgSimC$ and $localSim$, defined by the following equations.

$$avgSim(w, w') = \frac{1}{K}\frac{1}{K'}\sum_{i=1}^{K}\sum_{j=1}^{K'} s(v_s(w, i), v_s(w', j)) \tag{8}$$

where K and K' are the numbers of senses for w and w', $v_s(w, i)$ is the vector of the ith sense of word w, and $s(v_s(w, i), v_s(w', j))$ is the similarity measure between vectors $v_s(w, i)$ and $v_s(w', j)$. In our experiments, we apply cosine similarity as s.

$AvgSimC$ and $localSim$ can be computed when we have the context of the words.

$$avgSimC(w, w') = \frac{1}{K}\frac{1}{K'}\sum_{i=1}^{K}\sum_{j=1}^{K'} P(w, c, i)P(w', c', j)s(v_s(w, i), v_s(w', j)) \tag{9}$$

where $P(w, c, i)$ is the probability for word w to take the ith sense with context vector c.

$$localSim(w, w') = s(v_s(w, k), v_s(w', k')) \tag{10}$$

where $k = \text{argmax}_i P(w, c, i)$, $k' = \text{argmax}_{i'} P(w', c', i')$.

4.2.2 WordSim-353

WordSim-353 is a standard dataset for evaluating the quality of word vectors introduced by Finkelstein et al. (2001), which includes 353 pairs of nouns (without context). Each pair is presented with 13 to 16 human judgements on similarity and relatedness on a scale from 0 to 10. For example, pair (stock, market) gets the score of 8.08, while pair (stock, egg) only gains the score of 1.81.

In this dataset, since the context of words is not given, we can only compute the $avgSim$ for each pair of word to evaluate our method. The result is shown in Table 2.

4.2.3 SCWS

Stanford Contextual Word Similarity (SCWS) dataset proposed by Huang et al. (2012) is also a standard dataset to evaluate the performance of word embeddings quantitatively. It contains 2,003 pairs of words and the context they occur in.

Then as Neelakantan et al. (2015) did in their work, we also report the Spearman rank correlation between a model's output similarities and the human judgements. We also tried both random sampling and mean vector to get the representative vector for each pseudo multi-sense group. The result of our experiments are shown in Table 3.

Model	avgSim		
	original	random	mean
Huang et al. 50d	64.2	**65.1**	65.0
MSSG 50d	63.2	65.0	**65.1**
MSSG 300d	**70.9**	70.8	70.5
NP-MSSG 50d	62.4	64.0	**64.4**
NP-MSSG 300d	68.6	**69.1**	68.8

Table 2: Experimental result on WordSim-353 dataset (Spearman $\rho \times 100$). We apply both random choosing and mean vector to compute the representative vector for each group of pseudo multi-sense. Our method gains a slight improvement on all models except MSSG-300d.

Model	localSim			avgSim			avgSimC		
	original	random	mean	original	random	mean	original	random	mean
Huang et al.	26.1	**37.6**	36.9	62.8	61.4	62.9	65.7	65.9	66.1
MSSG 50d	49.2	52.4	**53.2**	64.2	64.9	64.8	66.9	67.0	67.2
MSSG 300d	57.3	62.1	**62.2**	67.2	67.3	67.2	69.3	69.1	69.4
NPMSSG50d	50.3	**55.5**	54.9	64.0	64.1	64.5	66.1	66.3	66.4
NPMSSG300d	59.8	**62.3**	62.2	67.3	67.3	67.4	69.1	68.9	69.2

Table 3: Experimental result on SCWS dataset (Spearman $\rho \times 100$). It shows that the elimination of pseudo multi-sense can significantly improves the performance of word embeddings with the metric *localSim*, while the performances of projected vectors on the metric *avgSim* and *avgSimC* are about the same as those of original vectors. In other words, the elimination of pseudo multi-sense improves the ability of representing a real sense of each sense vector locally.

4.3 Analogy

Analogy task is another method to evaluate the performance of word embeddings. In single-sense word embeddings, if the word A is similar to word B in the same sense as word C is similar to D, there should be an algebraic relationship $v(A) - v(B) = v(C) - v(D)$, where $v(A)$ is the vector of word A in the word embeddings (Mikolov et al., 2013a). Based on such relationship, we conduct the following experiment, which shows that our method is able to improve the quality of multi-sense word embeddings.

In order to compare the quality of different versions of word vectors, our experiment runs on the Semantic-Syntactic Word Relationship dataset, which contains five types of semantic questions and nine types of syntactic questions, as shown in Table 4, including 19544 such quadruples totally.

For each quadruple in the test dataset, we mark it as w_1, w_2, w_3, w_4. The relationship between w_1 and w_2 is similar to that between w_3 and w_4. In single-sense word embeddings, we just need to check whether $v(w_4)$ is the most similar vector to $v(w_1) - v(w_2) + v(w_3)$ among all the vectors, and apply the same procedure for w_1, w_2, w_3. For multi-sense word embeddings, we check whether there is a combination of senses $\{k_1, k_2, k_3, k_4\}$ so that $v_s(w_4, k_4)$ is the most similar vector to $v_s(w_1, k_1) - v(w_2, k_2) + v(w_3, k_3)$, where $v_s(w, k)$ is the vector of word w's k^{th} sense. What's more, since the equivalence of the two pairs, we also check by such procedure for $v_s(w_1, k_1), v_s(w_2, k_2), v_s(w_3, k_3)$. For every quadruple, once one of the requirements above is satisfied, we treat it as correct. We report the accuracy for each multi-sense vector space in Table 5.

Type of relationship	Word Pair 1		Word Pair 2	
Common capital city	Athens	Greece	Oslo	Norway
All capital cities	Astana	Kazakhstan	Harare	Zimbabwe
Currency	Angola	kwanza	Iran	rial
City-in-state	Chicago	Illinois	Stockton	California
Man-Woman	brother	sister	grandson	granddaughter
Adjective to adverb	apparent	apparently	rapid	rapidly
Opposite	possibly	impossibly	ethical	unethical
Comparative	great	greater	tough	tougher
Superlative	easy	easiest	lucky	luckiest
Present Participle	think	thinking	read	reading
Nationality adjective	Switzerland	Swiss	Cambodia	Cambodian
Past tense	walking	walked	swimming	swam
Plural nouns	mouse	mice	dollar	dollars
Plural verbs	work	works	speak	speaks

Table 4: Sample quadruple instances in analogy testing dataset. The relations are divided into 5 semantic types and 9 syntactic types.

Model	Semantic			Syntactic		
	original	random	mean	original	random	mean
Huang et al.	52.8	**53.5**	53.4	53.5	**56.1**	55.9
MSSG 50d	75.8	**77.5**	77.4	85.2	87.9	**88.0**
MSSG 300d	92.0	92.8	**93.1**	93.3	94.1	**94.5**
NPMSSG 50d	74.6	75.4	**75.6**	80.7	82.1	**82.3**
NPMSSG 300d	83.9	85.7	**85.9**	89.0	**90.2**	90.1

Table 5: Test result for analogy task. We also apply both random choosing and mean vector to get the representative vector for each pseudo multi-sense group. It shows that our improved vectors perform better on this task.

Overall, our detection and elimination of pseudo multi-sense on word embeddings reach higher performance on the nearest neighbor, word similarity and analogy task.

5 Conclusion and future work

In this paper, we introduced the concept of *pseudo multi-sense*, which is the word embedding models often embed one meaning to multiple senses, to describe the common problem in multi-sense word embeddings. Then we proposed a method based on both domains and semantic relations to detect such cases. What's more, we trained a global transition matrix based on the detected pseudo multi-sense from the given word embeddings, which is used to eliminate the distance between senses actually have the same meaning. The evaluation of our pseudo multi-sense eliminated vector showed that detecting and eliminating pseudo multi-sense significantly improved the ability for each vector in the word embeddings to represent for an exact meaning. We suggest that the following research directions could be considered.

- For the detection of pseudo multi-sense, taking syntactic information and other information we have or we can extract from corpus into account is a reasonable idea to improve the performance.

- Involve the pseudo multi-sense detection and elimination into the neural network structure, so that the learnt word embeddings could have higher quality than those learnt by existing methods without consideration of pseudo multi-sense.

- Though we have gained an improvement on experiments, we don't have a deep understanding about the reason that why elimination of pseudo multi-sense works well and why pseudo multi-sense cases are ubiquitous in all kinds of word embeddings. In future work, we could focus on finding a reasonable explanation of the fact.

References

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. 2003. A neural probabilistic language model. *journal of machine learning research*, 3(Feb):1137–1155.

Luisa Bentivogli, Pamela Forner, Bernardo Magnini, and Emanuele Pianta. 2004. Revising the wordnet domains hierarchy: semantics, coverage and balancing. In *Proceedings of the Workshop on Multilingual Linguistic Ressources*, pages 101–108. Association for Computational Linguistics.

Xinxiong Chen, Zhiyuan Liu, and Maosong Sun. 2014. A unified model for word sense representation and disambiguation. In *EMNLP*, pages 1025–1035. Citeseer.

Jianpeng Cheng and Dimitri Kartsaklis. 2015. Syntax-aware multi-sense word embeddings for deep compositional models of meaning. *arXiv preprint arXiv:1508.02354*.

Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*, pages 160–167. ACM.

Lev Finkelstein, Evgeniy Gabrilovich, Yossi Matias, Ehud Rivlin, Zach Solan, Gadi Wolfman, and Eytan Ruppin. 2001. Placing search in context: The concept revisited. In *Proceedings of the 10th international conference on World Wide Web*, pages 406–414. ACM.

Aitor González, German Rigau, and Mauro Castillo. 2012. A graph-based method to improve wordnet domains. In *International Conference on Intelligent Text Processing and Computational Linguistics*, pages 17–28. Springer.

Eric H Huang, Richard Socher, Christopher D Manning, and Andrew Y Ng. 2012. Improving word representations via global context and multiple word prototypes. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 873–882. Association for Computational Linguistics.

Jiwei Li and Dan Jurafsky. 2015. Do multi-sense embeddings improve natural language understanding? *arXiv preprint arXiv:1506.01070*.

Yang Liu, Zhiyuan Liu, Tat-Seng Chua, and Maosong Sun. 2015. Topical word embeddings. In *AAAI*, pages 2418–2424.

Bernardo Magnini and Gabriela Cavaglia. 2000. Integrating subject field codes into wordnet. In *LREC*.

Tomas Mikolov, Martin Karafiát, Lukas Burget, Jan Cernockỳ, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. In *Interspeech*, volume 2, page 3.

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, Quoc V Le, and Ilya Sutskever. 2013b. Exploiting similarities among languages for machine translation. *arXiv preprint arXiv:1309.4168*.

Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. 2013c. Linguistic regularities in continuous space word representations. In *HLT-NAACL*, volume 13, pages 746–751.

George A Miller. 1995. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.

Andriy Mnih and Geoffrey Hinton. 2007. Three new graphical models for statistical language modelling. In *Proceedings of the 24th international conference on Machine learning*, pages 641–648. ACM.

Andriy Mnih and Koray Kavukcuoglu. 2013. Learning word embeddings efficiently with noise-contrastive estimation. In *Advances in Neural Information Processing Systems*, pages 2265–2273.

Frederic Morin and Yoshua Bengio. 2005. Hierarchical probabilistic neural network language model. In *Aistats*, volume 5, pages 246–252. Citeseer.

Arvind Neelakantan, Jeevan Shankar, Alexandre Passos, and Andrew McCallum. 2015. Efficient non-parametric estimation of multiple embeddings per word in vector space. *arXiv preprint arXiv:1504.06654*.

Luis Nieto Pina and Richard Johansson. 2014. A simple and efficient method to generate word sense representations. *arXiv preprint arXiv:1412.6045*.

Joseph Reisinger and Raymond J Mooney. 2010. Multi-prototype vector-space models of word meaning. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 109–117. Association for Computational Linguistics.