LSTM Language Model for Hypernym Discovery

Hayato Hashimoto and Shinsuke Mori

Kyoto University, Yoshida Honmachi, Sakyo-ku, Kyoto 606-8501, Japan

Abstract. We introduce a new simple method in hypernym discovery task based on the distributional inclusion hypothesis. Our method utilizes LSTM language models to discover possible alternative words in the context of the given hyponym word and find the hypernym words within it. We implemented our method and tested it on a well-known hypernym discovery task, SemEval 2018 task 9. The experimental results showed that it achieved the state-of-the-art in comparison with existing unsupervised methods.

Keywords: hypernym discovery · hypernymy detection · hypernym-hyponym relationship · lexical entailment · is-a relationship · unsupervised learning · distributional inclusion hypothesis · distributional informativeness hypothesis · relative information content · recurrent neural network · LSTM · language model · contextual BLSTM

1 Introduction

Hypernym-hyponym relationship, also known as *is-a* relationship, is essential to model world knowledge for artificial intelligence. Several benchmark tasks [2, 3, 21, 25, 26] have been proposed to evaluate hypernymy detection methods. A hypernym detection task is a binary classification distinguishing word pairs v and w of is-a relations, or $v \prec w$ in short, from other word pairs. This task setting is, however, sometimes criticized that it tends to give high scores to models memorizing the words which frequently appear as hypernyms [22]. Additionally, the best models tend to be inconsistent among datasets, for example, figures reported by Chang et al. [7], for they depend on negative examples contained in the datasets. For example, some dataset contains random word pairs, and other datasets contain word pairs of *has-a* relationships as a negative example.

Recently, a new framework for evaluating hypernymy detection models is proposed, namely *hypernym discovery* [9]. Hypernym discovery is the task of finding the appropriate hypernyms for a given input word or phrase from the whole vocabulary of the corpus. This formulation enables us to test a system in settings more alike to real-world applications. Thus we focused on a domain-specific task of well-known SemEval 2018 shared task [6] in our experiments.

In the background explained in section 2, we came up with an idea that simple and straightforward approach based on alternativity of a word in its contexts may perform better than existing methods attempting to represent the sense of a word by a single vector. We utilize LSTM [12] language models (LMs) to calculate the word alternativity of w to v as the measure of being $v \prec w$, and find the hypernyms of v within words which have high alternativity to v.

We implemented our idea and tested it in SemEval 2018 shared task. In the subsequent sections, we first describe background and our method in detail and then evaluate it with its experimental results.

2 Background

The distributional inclusion hypothesis (DIH) [10] is a theoretical foundation of unsupervised learning of hypernymy relations, extending the distributional hypothesis [11]. Several measures [8, 14, 15, 27] for the word co-occurrence frequency or PPMI matrix [5] have been proposed in accord with this hypothesis.

The distributional informativeness hypothesis [20] is another theoretical foundation of unsupervised learning inspired by DIH. They hypothesize that a context of a word is more informative than the context of its hypernym, in the sense of the information theory.

Prediction-based approach for word semantics has been made in recent research: Baroni et al. [1] showed that vectors trained with Skip-Gram prediction task perform better in estimating word relatedness and resolving word analogies than count-based methods. For hypernymy, Chang et al. [7] proposed distributional inclusion vector embedding (DIVE), a model specially designed to capture the hypernymy-hyponym relations through Skip-Gram predictions, providing a firm baseline for hypernym discovery task.

3 Hypernymy Measure based on a Language Model

3.1 Definition of the Hypernymy Relationship

Adopting one of the definitions of is-a relationship proposed by Brachman [4], we define the hypernymy as the following.

Definition 1. *Let v_i be one of the senses of the term v and $\mathrm{Set}(v_i)$ denotes the set of the entities, the concepts, or the collections of things (for collective terms like "group") covered by the sense v_i. If and only if $\mathrm{Set}(v_i) \subset \mathrm{Set}(w_i)$, then "$w$ **is a hypernym of** v" ($v \prec w$).*

Hereafter, we assume a word has only one sense and equate the word sense w_i with its term w for simplicity.

3.2 Hypernyms are More Likely to Appear in the Context where their Hyponyms Appear

DIH is a statement that if $v \prec w$, then all syntactical properties of w also hold for v and vice versa. This hypothesis is based on the assumption that a hypernym can substitute for its hyponyms by definition.

But why a hypernym can substitute for its hyponym? Let us consider the case that an author wants to write a sentence about some real-world entities E and needs to determine the wording for E. Then, it is reasonable to assume that (s)he chooses the term v so that the sense of v covers E, and v has an appropriate level of abstraction and matches the writing style. By the definition 1, it automatically holds that w covers E when v covers E if $v \prec w$. Therefore, (s)he can choose w instead of v in context where v is appropriate, provided that w has an appropriate generality and matches the style.

Now let us consider the probability of occurrence of a certain word v in a given context c based on this argument. First we assume that the author mentions entities E in a context c with probability $P(c, E)$, which derives the conditional probability (in theory):

$$P(E \mid c) = \frac{P(c, E)}{\int_{E \in \Omega_E} \mathrm{d}P(c, E)},$$

where $\int_{E \in \Omega_E} \mathrm{d}P(c, E)$ denotes the marginal probability integrated by E. The conditional probability $P(v \mid c)$ can be modeled by the wording process as follows:

$$P(v \mid c) = P_{\mathrm{A}}(v \mid c) \cdot P_{\mathrm{S}}(v \mid c) \cdot P_{E \sim P(E|c)}(E \subset \mathrm{Set}(v)),$$

where $P_{E \sim P(E|c)}(E \subset \mathrm{Set}(v))$ denotes the probability that $E \subset Set(v)$ holds for the set of entities E drawn from the distribution $P(E \mid c)$, $P_{\mathrm{A}}(v \mid c)$ denotes the probability of the word in the context c being at the same level of abstraction as v, and $P_S(v \mid c)$ denotes the probability of the word in the context c matches the writing style to the same extent as v.

By the definition 1, if $v \prec w$ then $\mathrm{Set}(v) \subset \mathrm{Set}(w)$, and therefore, $P_{E \sim P(E|c)}(E \subset \mathrm{Set}(v)) \leq P_{E \sim P(E|c)}(E \subset \mathrm{Set}(w))$. Ignoring the writing style, if $P_{\mathrm{A}}(w \mid c)$ is constant or at least

$$\frac{P_{\mathrm{A}}(v \mid c)}{P_{\mathrm{A}}(w \mid c)} \leq \frac{P_{E \sim P(E|c)}(E \subset \mathrm{Set}(w))}{P_{E \sim P(E|c)}(E \subset \mathrm{Set}(v))}, \tag{1}$$

then it is concluded that $P(v \mid c) \leq P(w \mid c)$ for any context c. This result implies that a hypernym is more frequent than its hyponym. It also implies that $P(v, v_c) \leq P(w, v_c)$ if $v \prec w$ where $P(v, v_c)$ denotes the joint probability that the words v and v_c co-occur in the same bag-of-words, the same document, or the same latent topic, or there are instances of a certain type of syntactic dependency between v and v_c. As a corollary, it rationalize the estimation of hypernymy by calculating Clarke's degree of entailment (CDE) [8] or L_1 norm of the vectors from the frequency matrix of word co-occurrences [7] .

3.3 Filling Blanks by Language Models to Discover Hypernyms

We propose to use fitness to the *concrete context* which considers the whole sentence instead of bag-of-words or syntactic dependency relations.

Definition 2. ***concrete context*** *of v is the text blanked at the target word v.*

For example, "Antiviral drugs are _____ in treating influenza." is a concrete context of $v =$ "effective". For all w in the vocabulary V, we proposed to calculate the averaged fitness to the concrete contexts of v, which we call the word alternativity of w to v. The alternativity cannot be measured by counting word occurrences since the concrete context is almost unique. Instead, we propose to use an LM to calculate the probability of alternative words filling in the blank.

We propose to define a measure for word alternativity of w to v, namely **LM-Measure**, as the following:

Definition 3. **X-*LM-Measure*** $(v \prec w)$ *is the difference of log-likelihood or relative information content averaged over all concrete contexts* $C[v]$ *of* v*:*

$$\textbf{LM-Measure}\,(v \prec w) = \mathop{\mathbb{E}}_{c \in C[v]} \left[\log P_{\mathbf{LM}}(w \mid c) - \log P_{\mathbf{LM}}(v \mid c)\right],$$

where $P_{\mathbf{LM}}$ *is a word occurring probability according to the* **X** *language model.*

3.4 Contextual Language Models

The conventional LMs predict the word only from the preceding words. This model often mistakes the part-of-speech (POS), for example in section 3.1, the LSTM LM predicted “not,” “generally,” or “the” to fill in the blank in the above example. Thus we adopted to use the contextual language model (cLM) where both the preceding and the succeeding words are provided. A cLM trained by the method described in the next section predicted “effective,” “useful,” or “recommended” belonging to the proper POSs to fill in the blank in the same example.

3.5 Learning Contextual Bidirectional LSTM

Contextual BLSTM [18] or cBLSTM is a cLM which consists of two LSTMs which consumes word sequences forward and backward (Fig. 1).

We calculated the logit vector $\boldsymbol{h} = \boldsymbol{h}_{f,k-1} + \boldsymbol{h}_{b,k+1}$ using $\boldsymbol{h}_{f,k-1}$ and $\boldsymbol{h}_{b,k+1}$ obtained from the $(k-1)$-th and the $(k+1)$-th output of the forward and backward LSTMs, respectively. We tied the parameters of the linear layer with the embedding layer following [13]. We trained it by minimizing the categorical cross entropy loss of the predicted probability distribution.

3.6 Fast Calculation of LM-Measure

For LMs or cLMs outputting word probability using softmax function to normalize the logit vector $\boldsymbol{h}$, the relative information content $\log P_{\mathbf{LM}}(w|c) - \log P_{\mathbf{LM}}(v|c)$ can be calculated by

$$\begin{aligned}\log[\mathrm{softmax}(\boldsymbol{h})]_v - \log[\mathrm{softmax}(\boldsymbol{h})]_w &= \log \frac{\exp(h_v)}{\sum_{i \in V} \exp(h_i)} - \log \frac{\exp(h_w)}{\sum_{i \in V} \exp(h_i)} \\ &= h_v - h_w,\end{aligned}$$

where the subscript h_v denotes the scalar element of the vector $\boldsymbol{h}$ at the index corresponding to the word v. In our experiments, we obtained **LM-Measure** for all pairs of words in one path using the following procedure. First, we calculated $h_v - h_w$ for all w in the most common $|V'|$ words for all v in the corpus in order of appearance. The sum of the relative information content was stored in a matrix $W_{\text{Measure}} \in \mathbb{R}^{|V| \times |V'|}$. Then, we divided every row by the corpus frequency of the corresponding word. We ran this path as fast as running one epoch of the training of the cBLSTM.

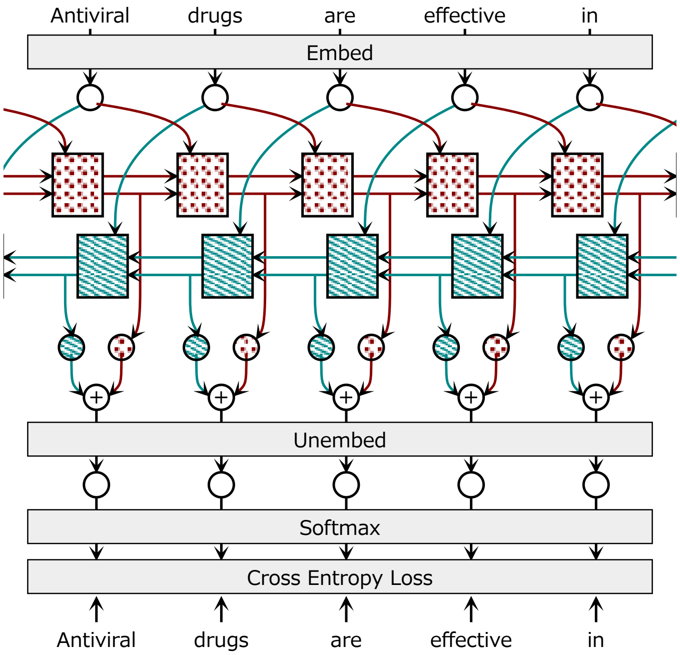

Fig. 1. cBLSTM combines forward and backward LSTMs to predict a word in a given context. The square boxes in the middle stand for LSTM cell and circles stand for hidden vectors.

Table 1. Hyperparameters used to train the baseline models

word2vec	
Window Size	8
# of negative samples	25
cbow	1
Downsampling threashold	10^{-4}
Training epochs	15
DIVE	
# of negative samples	30
Inclusion Shift	Yes
Window size	5
PMI filter size	5
Embedding size	100
Training epochs	15

4 Evaluation

4.1 Training Corpus

We retrieved the pre-tokenized corpus from the competition website of SemEval 2018 [1]. The task consists of five subtasks: three for general domains for different languages (1A, 1B, and 1C) and two for specific domains (2A–medical and 2B–music). We conducted experiments only on 2A and 2B, since it appears to be more practical settings for unsupervised learning while dataset is relatively small. The corpus originates from several in-domain sources. We split the text into the first 1,000 lines for validation and rest for training corpus. We limited the vocabulary to the most common 80,000 words in the corpus, and all out-of-vocabulary words were rendered to `<unk>`.

Randomly taken 1%, 10%, and 100% sentences from the training corpus are used to train the baseline and proposed models to compare the degree of dependence on the size of the corpus.

4.2 Model Setups

Baselines In the results section, table 2 cites the results of unsupervised methods summarized by Camacho-Collados et al. [6]. ADAPT [16] calculates cosine similarity of word embedding vectors learned by skip-gram negative sampling with the given corpus. APSyn [19] is a symmetric measure to calculate the similarity of two words using the rank of shared context in the PPMI matrix. SLQS [20] is an asymmetric measure to calculate the differences in generality between two words by exploiting the distributional informativeness hypothesis.

In addition to these methods, we set two unsupervised baselines, namely **WΔS** and **ClogP**.

WΔS: We adopted the **WΔS** hypernymy measure from [7] as a baseline. **WΔS** is the cosine similarity of the `word2vec` [17] vectors multiplied by the generality signal $\Delta S(v \prec w) = \|w\|_1 - \|v\|_1$ of DIVE.

[1] https://competitions.codalab.org/competitions/17119

We trained the `word2vec` model by the Mikolov's code[2] and the DIVE model by the code provided by the authors[3]. We searched for the best number of dimensions d of `word2vec` vectors, but other hyper-parameters are fixed (Table 1).
ClogP: Since **LM-Measure** gives high score to frequent words, we set the following baseline to see if **LM-Measure** do more than selecting similar and common words:

$$\textbf{ClogP}\,[\alpha]\,(v \prec w) = \cos\left(\boldsymbol{v}, \boldsymbol{w}\right) + \alpha \cdot \log P(w),$$

where $\boldsymbol{v}$ and $\boldsymbol{w}$ are `word2vec` representations of v and w, respectively, and α is a hyper-parameter and $P(w)$ is the frequency of w in the corpus divided by the total number of words.

Training Language Models for LM-Measure We trained LSTM and cBLSTM using step annealing for 20 epochs. Firstly we split the corpus into n parts and the first m words of each part are feed into the LSTM or cBLSTM in a batch. The hidden states of the LSTM or cBLSTM are passed to the next batch, which processes the next m words [4]. We calculated the hypernymy measure as described in section 3 using $|V'| = 20000$.

LM-Measure Variations We added the following variations to the proposed **LM-Measure** model.
Label smoothing: To ensure that cBLSTM predicts a large variety of words, we adopted the label smoothing technique [23, 24]. We imposed 0.9 and $0.1/(|V| - 1)$ for correct and incorrect predictions instead of 1 and 0 as a supervision signal when calculating cross entropy during training.
Discounting word frequency: In general, a cLM gives a high probability to frequent words. In the presence of validation dataset, we can tune the system to cancel this effect by adequately discounting the measure by the log-frequency of w. We calculated **LM-Measure** $(v \prec w) - \alpha \cdot \log P(w)$ as a hypernymy measure, where α is a hyper-parameter.
POS filtering: In this variation, we filtered out the hypernym candidates in case that their POS does not match the given hyponym word. We used the WordNet database as a POS dictionary to find if any synset of the candidate hypernym be a noun, for all hyponym words are nouns in the dataset used in the experiment.

4.3 Validation and Test Dataset of Hypernym-Hyponym Pairs

We obtained the validation and test dataset from competition website for SemEval 2018. We used dataset originally provided for training supervised systems as the validation dataset. Validation and test dataset contains 500 hyponyms $V_{dataset} = \{v_1, v_2, ..., v_{500}\}$ and the corresponding lists of hypernyms $W_{v_1}, W_{v_2}, ..., W_{v_{500}}$ such that

[2] https://github.com/tmikolov/word2vec

[3] https://github.com/iesl/Distributional-Inclusion-Vector-Embedding

[4] This feeding process disturbs the word order for backward LSTM of cBLSTM once in m words, but it enables efficient learning since it can make $n \times m$ predictions at once. The dimensions of the embedding vector and the hidden state is set to 300.

$\forall w \in W_{v_i}(v_i \prec w)$, and each word v_i is labeled with either “Concept” or “Entity.” We excluded entities and left concepts because unsupervised learning is not suitable for finding hypernyms of entities since there is only a little number of possible hypernyms, being more like a classification problem.

4.4 Evaluation Details

As described by Camacho-Collados et al. [6], we estimated fifteen hypernym candidates $w_{v,1}, w_{v,2}, ..., w_{v,15}$ for each given word $v \in V_{dataset}$ and calculated mean average precision (MAP), mean reciprocal rank (MRR), and precision at $k = 1$, 5, and 15 (P@k) by comparing them with the corresponding hypernyms W_v in the dataset. We used the implementation provided by Camacho-Collados et al. [6] to calculate these metrics. We excluded 180 stop words and the given word itself from the candidates. The stop word list is basically the same as the one used in the DIVE.

For input phrases more than one word, we calculated the score of the last word. We also conducted an experiment calculating hypernym scores using the averaged vectors or the averaged **LM-Measure** for all words appeared in a phrase, but the performance was inferior.

We tuned α and the dimension d of `word2vec` vectors by choosing the ones maximizing MAP of the validation pairs. We searched over (100, 300, 1000) for d and (1/5, 1/10, ..., 1/30, 0) for α.

4.5 Results and Discussion

Table 2 shows the result of our experiments. Fistly, we observed that even the **ClogP** baseline surpasses the result reported in literature [6]. In hypernym discovery tasks, it seems that it is effective to narrow down the large vocabulary by selecting more general and similar words, and the word frequency is a reliable index to measure the generality of a word.

Secondly, we observed that **LM-Measure** worked without tuning discounting hyperparameter. We suspect that contributions to the hypernymy measure of the similarity and the generality are already balanced by the information content approach.

Thirdly, we observed that **LM-Measure** surpasses **ClogP** baseline, especially at P@5 and P@15. These figures imply that it picked up a wider variety of correct hypernyms than **ClogP** baseline. To substantiate this point, we listed the correct hypernyms which appear in the top 5 hypernym candidates list obtained by one method but do not appear in the top 15 hypernym candidates obtained by the other method (Table 3) and examined the hypernym candidates of “influenza” (Table 5 .). It seems that **ClogP** preferred co-occurring related concepts and failed to predict “disease” since it did not co-occur with disease names since the authors avoid redundancy, while **LM-Measure** succeed to predict “disease.”

We tested ability of **LM-Measure** to retrieve hypernyms from frequent (and hence presumably generic) words by calculating the average precision for each hypernyms and showing its dependency to the word frequency. Fig. 2 shows the result of this investigation. The figure shows that **LM-Measure** estimates hypernym candidates better

Table 2. Results of the subtasks 2A and 2B of the SemEval 2018 task 9 Bold figures indicate the best score in the same corpus size. Note that Anu is the model trained with both the corpus and the WordNet.

Subtask	2A - medical domain					2B - music domain				
Performance metric	MAP	MRR	P@1	P@5	P@15	MAP	MRR	P@1	P@5	P@15
Training data used: 1%										
ClogP	10.03	29.99	29.20	8.87	7.03	4.03	11.67	9.78	3.47	2.66
cBLSTM-LM-Measure	14.54	39.30	**36.80**	13.91	10.26	**6.85**	**19.87**	**15.92**	6.45	**5.10**
+ discount	**15.02**	**39.81**	**36.80**	**14.52**	**10.49**	6.82	19.70	15.64	**6.56**	5.08
+ smoothing	14.13	38.58	36.00	13.53	9.91	6.68	19.12	15.36	6.49	4.86
+ smoothing + discount	14.30	38.58	35.60	13.85	9.98	6.68	19.16	15.36	6.49	4.84
Training data used: 10%										
ClogP	14.66	40.31	38.60	13.27	10.47	6.99	20.26	16.76	6.54	5.03
cBLSTM-LM-Measure	16.77	43.81	**39.20**	16.52	12.03	**8.40**	22.28	16.48	8.04	**6.55**
+ discount	**16.97**	**43.87**	38.40	**16.74**	**12.11**	**8.40**	22.28	16.48	8.04	**6.55**
+ smoothing	16.54	42.84	38.40	16.06	11.87	8.36	**23.11**	**17.60**	**8.16**	6.25
+ both	16.62	43.23	38.60	16.18	11.84	8.36	**23.11**	**17.60**	**8.16**	6.25
Training data used: 100%										
ClogP	13.96	39.14	36.60	12.89	10.31	7.19	20.31	16.20	7.37	5.45
W∆S	3.20	9.21	6.00	3.32	2.45	2.98	8.64	5.03	3.05	2.36
LSTM-LM-Measure	12.62	33.19	25.40	13.09	9.01	5.11	14.03	9.22	5.40	4.09
+ POS filtering	14.49	38.95	33.00	14.55	10.07	5.92	16.04	10.06	6.14	4.70
+ POS filtering + discount	14.44	38.86	32.80	14.59	9.95	5.92	16.04	10.06	6.14	4.70
cBLSTM-LM-Measure	16.81	42.64	37.40	16.78	12.22	8.84	24.17	18.44	8.81	6.58
+ discount	16.91	42.77	37.40	16.98	12.28	8.84	24.17	18.44	8.81	6.58
+ smoothing	16.75	42.82	**37.60**	16.68	12.14	8.17	21.83	16.76	7.80	6.61
+ smoothing + discount	16.75	42.82	**37.60**	16.68	12.14	8.17	21.83	16.76	7.80	6.61
+ POS filtering	16.93	43.04	**37.60**	16.91	12.36	**8.90**	**24.24**	**18.44**	**8.81**	**6.64**
+ POS filtering + discount	**17.02**	**43.18**	**37.60**	**17.03**	**12.37**	**8.90**	**24.24**	**18.44**	**8.81**	**6.64**
Results of unsupervised methods for Concepts summerized by Camacho-Collados et al. [6]										
ADAPT [16]	8.13	20.56	-	8.32	-	1.88	5.34	-	1.89	-
Anu	7.05	17.51	-	7.29	-	*10.68*	*27.13*	-	*10.84*	-
(Team 13)	2.55	7.19	-	2.52	-	4.83	14.33	-	4.51	-
balAPInc [14]	0.91	2.10	-	1.08	-	1.44	3.65	-	1.58	-
APSyn [19]	0.65	1.43	-	0.72	-	1.13	2.55	-	1.30	-
SLQS [20]	0.29	0.66	-	0.33	-	0.64	1.25	-	0.65	-

Table 3. Hypernyms found unique to each method LM-Measure can find generic words as hypernyms compared to ClogP.

LM-Measure		**ClogP**	
hypernym	count	hypernym	count
disease	83	enzyme	2
drug	8	anemia	2
pain	4	pigment	1
disorder	3	neoplasm	1
(other 15 words)	1 or 2	(other 20 words)	1

Table 4. Hypernyms found unique to each language model Contextual language model can find more specific words as hypernyms compared to a non-contextual language model.

cBLSTM		**LSTM**	
hypernym	count	hypernym	count
drug	2	disease	2
allergen	1	blood	1
constipation	1	fluid	1
corticosteroid	1	ion	1
(other 22 words)	1		

Table 5. Hypernym candidates for "influenza" The proposed **LM-Measure** successfully estimates the word "disease" and "infection" as hypernyms while **ClogP** wrongly estimates co-occurring words ("outbreak", "seasonal") and synonyms ("flu").

LM-Measure	**ClogP**
Influenza respiratory virus RSV HIV viral cancer pandemic <eos> hepatitis disease H5N1 H1N1 adenovirus lung H3N2 infection infectious human	Influenza H1N1 pandemic RSV ILI pH1N1 viruses outbreak SARI pdm09 ARI outbreaks virus H3N2 H5N1 viral seasonal respiratory flu

if they have large corpus frequency, and for frequent hypernyms **ClogP** does not match the LM-Measure results even if the large α is set to prioritize the frequent words. It seems that if there is larger difference in generality between two words, `word2vec` vectors no longer show high cosine similarity.

In this investigation, we used the following defition of average precision AP of the hypernym w:

$$AP(w) = \frac{1}{|V_w|} \sum_{v \in V_w} \frac{\sum_{k=1}^{r(v,w)} \mathbf{1}_{W_v}(w_{v,k})}{r(v,w)},$$

where $V_w \subset V_{dataset}$ is the set of hyponyms of w in the test dataset, $w_{v,k}$ is the k-th hypernym candidate of v, r is the rank of w (i.e. $w_{v,r(v,w)} = w$), and

$$\mathbf{1}_{W_v}(w) = \begin{cases} 1 & (w \in W_v, \text{ i.e. } v \prec w) \\ 0 & (\text{otherwise}) \end{cases}.$$

One limitation of **LM-Measure** is that it sometimes considers the hyponym as a hypernym. In Table 5 , **LM-Measure** wrongly estimates "H5N1" being a hypernym of "influenza," while it is, in fact, a hyponym of "influenza." It may reflects the fact that medical research papers preferably describe this disease at the subtype level to retain academic detailedness, and the inequality assumption (1) does not hold in this case.

Lastly, POS filtering shows large improvement in **LSTM-LM-Measure** but small improvement in **cBLSTM-LM-Measure** as expected. The results also shows that POS filtering alone is not enough to fill the gap between **LSTM-LM-Measure** and **cBLSTM-LM-Measure**. Table 4 shows the correct hypernyms appear in the top 5 candidates obtained by **LM-Measure** + POS filtering using one language model but do not appear in the top 15 hypernym candidates obtained by **LM-Measure** + POS filtering using the other language model.

5 Conclusion and Future Work

By taking the straightforward approach utilizing the distributional inclusion hypothesis, our proposed method achieved state-of-the-art in unsupervised hypernym discovery tasks.

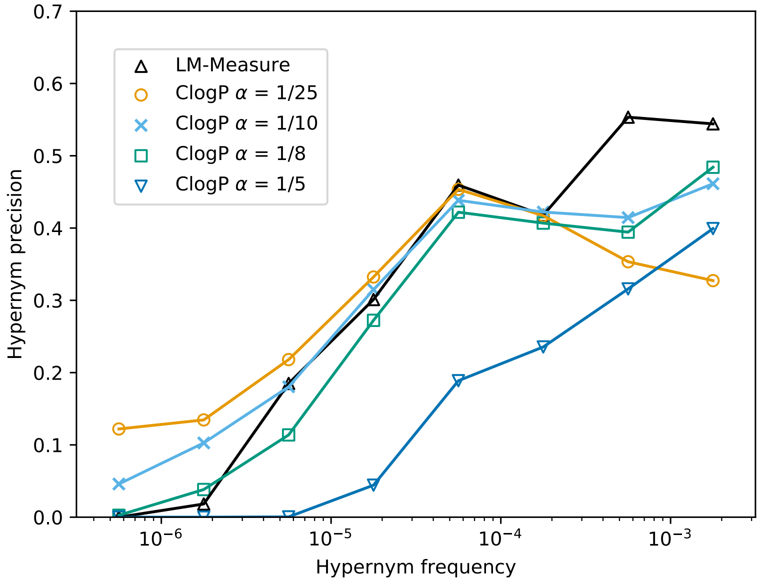

Fig. 2. LM-Measure estimates hypernym candidates better if they have large corpus frequency. Hypernym precision of the proposed **cBLSTM-Measure** and the **ClogB** baseline using different hyperparameters α were compared by the subtask 2A of the SemEval 2018 task 9. The hypernyms were divided into eight bins according to their log frequency (horizontal axis) and mean precision (vertical axis) were calculated for each bin.

Our method can be extended from words to phrases by replacing a blank with n consecutive blanks when calculating **LM-Measure**. Another direction of extension is to use attentional models as the cLM. The Transformer LM [24] can be trained as a cLM by masking.

6 Acknowledgement

This research was partially supported by the Cooperative Intelligence Joint Research Chair with Honda Research Institute Japan Co., Ltd.

We thank Keisuke Shirai for indicating label smoothing technique, and anonymous reviewers of NAACL-HLT 2019 for comments that greatly improved the manuscript.

Bibliography

[1] Baroni, M., Dinu, G., Kruszewski, G.: Don't count, predict! a systematic comparison of context-counting vs. context-predicting semantic vectors. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 238–247. Association for Computational Linguistics (2014). https://doi.org/10.3115/v1/P14-1023

[2] Baroni, M., Lenci, A.: How we blessed distributional semantic evaluation. In: Proceedings of the Workshop on Geometrical Models of Natural Language Semantics (GEMS '11). pp. 1–10. Association for Computational Linguistics, Stroudsburg, PA, USA (2011), http://aclweb.org/anthology/W11-2501

[3] Benotto, G.: Distributional models for semantic relations: A study on hyponymy and antonymy. PhD Thesis, University of Pisa (2015), https://etd.adm.unipi.it/theses/available/etd-04302015-171419/

[4] Brachman, R.J.: What is-a is and isn't: An analysis of taxonomic links in semantic networks. Computer **16**(10), 30–36 (Oct 1983). https://doi.org/10.1109/MC.1983.1654194

[5] Bullinaria, J.A., Levy, J.P.: Extracting semantic representations from word co-occurrence statistics: A computational study. Behavior research methods **39**(3), 510–526 (2007). https://doi.org/10.3758/BF03193020

[6] Camacho-Collados, J., Delli Bovi, C., Espinosa Anke, L., Oramas, S., Pasini, T., Santus, E., Shwartz, V., Navigli, R., Saggion, H.: Semeval-2018 task 9: Hypernym discovery. In: Proceedings of the 12th International Workshop on Semantic Evaluation. pp. 712–724. Association for Computational Linguistics (2018), http://aclweb.org/anthology/S18-1115

[7] Chang, H.S., Wang, Z., Vilnis, L., McCallum, A.: Distributional inclusion vector embedding for unsupervised hypernymy detection. In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers). pp. 485–495. Association for Computational Linguistics, New Orleans, Louisiana (June 2018), http://aclweb.org/anthology/N18-1045

[8] Clarke, D.: Context-theoretic semantics for natural language: An overview. In: Proceedings of the Workshop on Geometrical Models of Natural Language Semantics (GEMS '09). pp. 112–119. Association for Computational Linguistics, Stroudsburg, PA, USA (2009), http://aclweb.org/anthology/W09-0215

[9] Espinosa Anke, L., Camacho-Collados, J., Delli Bovi, C., Saggion, H.: Supervised distributional hypernym discovery via domain adaptation. In: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing. pp. 424–435. Association for Computational Linguistics (2016), http://aclweb.org/anthology/D16-1041

[10] Geffet, M., Dagan, I.: The distributional inclusion hypotheses and lexical entailment. In: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics. pp. 107–114. ACL '05, Association for Computational Linguistics, Stroudsburg, PA, USA (2005). https://doi.org/10.3115/1219840.1219854

[11] Harris, Z.S.: Distributional structure. Word **10**(2-3), 146–162 (1954). https://doi.org/10.1080/00437956.1954.11659520

[12] Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Computation **9**(8), 1735–1780 (Nov 1997). https://doi.org/10.1162/neco.1997.9.8.1735

[13] Inan, H., Khosravi, K., Socher, R.: Tying word vectors and word classifiers: A loss framework for language modeling. In 7th International Conference on Learning Representations (2017), https://arxiv.org/abs/1611.01462

[14] Kolterman, L., Dagan, I., Szpektor, I., Geffet, M.: Directional distributional similarity for lexical inference. Natural Language Engineering **16**(4), 359–389 (2010). https://doi.org/10.1017/S1351324910000124

[15] Lenci, A., Benotto, G.: Identifying hypernyms in distributional semantic spaces. In: Proceedings of the First Joint Conference on Lexical and Computational Semantics - Volume 1: Proceedings of the Main Conference and the Shared Task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation. pp. 75–79. SemEval '12, Association for Computational Linguistics, Stroudsburg, PA, USA (2012), http://aclweb.org/anthology/S12-1012

[16] Maldonado, A., Klubička, F.: Adapt at semeval-2018 task 9: Skip-gram word embeddings for unsupervised hypernym discovery in specialised corpora. In: Proceedings of The 12th International Workshop on Semantic Evaluation. pp. 924–927. Association for Computational Linguistics (2018), http://aclweb.org/anthology/S18-1151

[17] Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. In: Workshop Proceedings of the International Conference on Learning Representations (ICLR 2013) (2013), http://arxiv.org/abs/1301.3781

[18] Mousa, A., Schuller, B.: Contextual bidirectional long short-term memory recurrent neural network language models: A generative approach to sentiment analysis. In: Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers. pp. 1023–1032. Association for Computational Linguistics, Valencia, Spain (April 2017), http://aclweb.org/anthology/E17-1096

[19] Santus, E., Lenci, A., Chiu, T.S., Lu, Q., Huang, C.R.: What a nerd! beating students and vector cosine in the esl and toefl datasets. In: Chair), N.C.C., Choukri, K., Declerck, T., Goggi, S., Grobelnik, M., Maegaard, B., Mariani, J., Mazo, H., Moreno, A., Odijk, J., Piperidis, S. (eds.) Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016). European Language Resources Association (ELRA), Paris, France (may 2016), http://www.lrec-conf.org/proceedings/lrec2016/summaries/517.html

[20] Santus, E., Lenci, A., Lu, Q., Schulte im Walde, S.: Chasing hypernyms in vector spaces with entropy. In: Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, volume 2: Short Papers. pp. 38–42. Association for Computational Linguistics, Gothenburg, Sweden (April 2014), http://aclweb.org/anthology/E14-4008

[21] Santus, E., Yung, F., Lenci, A., Huang, C.R.: Evalution 1.0: an evolving semantic dataset for training and evaluation of distributional semantic models. In: Proceedings of the 4th Workshop on Linked Data in Linguistics: Resources and Applica-

tions. pp. 64–69. Association for Computational Linguistics, Beijing, China (July 2015), http://aclweb.org/anthology/W15-4208

[22] Shwartz, V., Santus, E., Schlechtweg, D.: Hypernyms under siege: Linguistically-motivated artillery for hypernymy detection. In: Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers. pp. 65–75. Association for Computational Linguistics, Valencia, Spain (April 2017), http://aclweb.org/anthology/E17-1007

[23] Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2016). https://doi.org/10.1109/CVPR.2016.308

[24] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L.u., Polosukhin, I.: Attention is all you need. In: Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (eds.) Advances in Neural Information Processing Systems 30, pp. 5998–6008. Curran Associates, Inc. (2017), http://papers.nips.cc/paper/7181-attention-is-all-you-need.pdf

[25] Vendrov, I., Kiros, R., Fidler, S., Urtasun, R.: Order-embeddings of images and language. In: Conference Proceedings of the International Conference on Learning Representations (ICLR 2016) (2016), https://arxiv.org/abs/1511.06361

[26] Weeds, J., Clarke, D., Reffin, J., Weir, D., Keller, B.: Learning to distinguish hypernyms and co-hyponyms. In: Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers. pp. 2249–2259. Dublin City University and Association for Computational Linguistics (2014), http://aclweb.org/anthology/C14-1212

[27] Weeds, J., Weir, D.: A general framework for distributional similarity. In: Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing (EMNLP '03). pp. 81–88. Association for Computational Linguistics, Stroudsburg, PA, USA (2003). https://doi.org/10.3115/1119355.1119366