English Out-of-Vocabulary Lexical Evaluation Task

Anonymous ACL submission

Abstract

Unlike previous unknown nouns tagging task (Curran, 2005) (Ciaramita and Johnson, 2003), this is the first attempt to focus on out-of-vocabulary(OOV) lexical evaluation tasks that does not require any prior knowledge. The OOV words are words that only appear in test samples. The goal of tasks is to provide solutions for OOV lexical classification and predication. The tasks require annotators to conclude the attributes of the OOV words based on their related contexts. Then, we utilize unsupervised word embedding methods such as Word2Vec(Mikolov et al., 2013) and Word2GM (Athiwaratkun and Wilson, 2017) to perform the baseline experiments on the categorical classification task and OOV words attribute prediction tasks.

1 Introduction

The evolution of modern English language brings new words in and eliminates old words out. Thus out-of-vocabulary (OOV) handling is an inevitable challenge among nearly all natural language processing topics: In cross-lingual translation, the quality of translation is heavily dependent on the identification of OOV words. In speech-to-text translation, detecting and modelling the OOV words can significantly improve the lexical completeness of the input. In semantic analysis, a majority of OOV words are proper names (PN) which are important for discovering the contextual concept relatedness. In social network sentiment analysis, new words are created every day on the Internet, thus the ability of handling OOV words can also contribute to the robustness of the model. Moreover, the concept of OOV handling is closely related with the emerging ”zero shot learning (ZSL)”(Zhang and Saligrama, 2015) in other machine learning studies.

Therefore, it is indispensable to investigate the metrics of evaluating the OOV classifications and predictions. Although in NLP study there are various semantic evaluation tasks for computational semantic analysis such as (McCarthy and Navigli, 2007), (Specia et al., 2012) and (Siddharthan, 2006), the majority of current evaluation tasks focus on word sense disambiguation and text simplification. (Dagan et al., 2006) describes their work to capture major semantic inferences across applications. Finding a valid substitution with the given context has been proven to be effective in question answering, abstract extracting, etc. The task proposed in this paper concentrates more on the dominant OOV semantic prediction. The main difference between the other semantic simplification tasks is that OOV words may not impede the performance in their tasks, while in our proposed task, the outcome is directly related with the OOV words.

In previous unknown nouns supersense tagging,(Curran, 2005) and (Ciaramita and Johnson, 2003), rule-based models were proposed within the scope of WordNet. The main limitations are:

- The models are heavily dependent on some particular features of the unknown words such as the part of speech tagging and chunking, morphological analysis and even gramatical relation extraction.
- Supersense method is based on WordNet hierarchy, called *lexicographer files*, which contains redundant prior knowledge.
- These models cannot handle words that do not exist in WordNet. However, the vocabulary of Wikipedia we used is ten times larger than that of WordNet.

The task proposed in this paper jumps out from the above limitations. First, both training and testing data we utilized are from Wikipedia, which require no prior knowledge. Second, the baseline experiments are purely unsupervised and are based on vector-space rather than predefined rules.

Word Sense Disambiguation(WSD) is to identify the meanings of ambiguous words. We apply WSD in our baseline experiments and compare it with non-WSD models.

2 Task Statement

OOV handling in natural language processing is still a sophisticated task due to several reasons. Lack of definition or explanation about the particular OOV word is one of the obstacles to accurately predict the meanings of OOV words. Another aspect is, newly created words usually have a comparably lower frequency than normal words, and sometimes they are eliminated when the minimal count is performed in preprocessing, Therefore, recognition of OOV highly depends on the quality of corpora. Given the fact that no corpus in use is the latest real-world corpus, it is essentially helpful to take the OOV classification and prediction capacity into consideration as a robust metric of NLP models. We hence propose our OOV classification and prediction tasks. Each task contains three parts, an OOV word, context and attributes. The given context is as close as a description to the OOV word and it can be either long or short regarding the rarity and difficulty of the OOV word. We assigned several attributes to each of the OOV word based on the corresponding context. The attributes can be the topic of the context, the potential or prior characteristics about the OOV word. Attributes are provided by independent annotators. We invited five individuals in USA, they are pursuing master or PhD degrees and are all fluent but non-native English speakers. Each context is concluded by three random annotators and we intersect the common attributes.

Another contribution in this paper is that we introduced a category classification task of OOV words with context. When we analyzed the rare words in Wikipedia English corpus, we found that many of the words are chemical and medical science terminologies with obscure contexts, which could not be comprehensively understood by human annotators. In order to make the proposed task more generic, we discarded those contexts, and hand-picked 5 categories: Greek mythology, locations, animals, plants and technology.

2.1 Data source

The corpus was collected from Wikipedia dumps 2017-10-20 version (https://dumps.wikimedia.org/backup-index.html). The raw XML format corpus contains various types of pre-defined categories, each category has an unambiguous structure tree with its children nodes representing the sub-categories, which is similar to WordNet (https://wordnet.princeton.edu/). Wikipedia category tree has much more nodes compared with WordNet. The majority of the selected OOV words actually do not exist in WordNet.

2.2 Data selection

Regarding the rules of OOV sampling, firstly words with both too high and too low occurrence were eliminated because we need to limit the data size and those eliminated words are usually less informative, which can be treated as stopwords. A reasonable data size is vital in terms of performance and speed optimization for both training and testing. Secondly, an OOV word is restricted to appear only in given context, and this is also our condition to retrieve such an OOV. An advantage to utilize Wikipedia as the data source is its sufficient entries satisfying the requirements.

After the occurrence filtering, there yet exists numerous OOV candidates. Thus we only select some particular but convictive categories. As mentioned above, according to statistical analysis, those five categories contain abundant OOV words and their given contexts are readable. Oppositely, in the category of medicine and chemistry, the context is not readily comprehensible for our annotators.

The last step for data selection is the attribute extraction. Each OOV word and its context is designated to three random annotators. After the individual conclusion, an attribute intersection is performed to acquire final results.

Examples:

Word: *arachis*

Context:

Arachis is a genus of about 70 species of annual and perennial ***flowering*** *plants in the* ***pea*** *family (****Fabaceae****), native to South America, and was recently assigned to the informal monophyletic Pterocarpus clade of the Dalbergieae.*

This OOV's category is plant, while the attributes are pea, flower and fabaceae, which can all be found in context.

Word: *winwebsec*

Context:

Winwebsec is a category of ***malware*** *that targets the users of Windows operating systems and produces fake claims as genuine anti-malware software, then demands payment to provide fixes to fictitious problems.*

This OOV's category is technology, while its attributes are malware, adware, spyware, and the context contains only malware.

The whole data set for the two tasks can be accessed from (https://github.com/hwangtamu/OOVLexical)

2.3 Task 1

The first task is to test if the positive examples are correctly classified and the negative examples are correctly excluded from their corresponding categories. More specifically, the ground truth is set as the higher level Wikipedia category names from which the entries were taken.

$$S_1 = \frac{1}{N}\sum_{i=1}^{N} \min R_i \tag{1}$$

where R_i represents rank of a correct prediction and N is the total number of test samples.

2.4 Task 2

The second task is to test if top K semantic predictions of the OOV words hit the human annotated attributes. Each OOV word can have up to 5 annotated share-weighted attributes, therefore we proposed a different scoring criterion than Task 1.

$$S_2 = \frac{1}{KN}\sum_{i=1}^{N} bool(\exists w_{ij} \in W_i \cap \hat{W}_i) \tag{2}$$

where W_i is the top K prediction set of the ith test sample and $\hat{W}_i$ is the annotated set of ith test sample.

We provide details of baseline experiments in next chapters, with the S_1 and S_2 scores reported.

3 Experiment

We used Word2Vec(Mikolov et al., 2013) and Word2GM(Athiwaratkun and Wilson, 2017) as baseline models of the tasks. The data set we utilized for training is a subset of the Wikipedia corpus with $\sim$ 37M tokens, and $\sim$ 340K unique tokens. The low frequency words (appear less than 5 times) are removed from the training set, leaving a vocabulary of $\sim$ 75K to out models.

3.1 Word2Vec

The skip-gram Word2Vec model can efficiently project a vocabulary of words into a finite dimensional vector space R^d by maximizing

$$P(c|w; v_{c_i}, v_{w_i}) = \frac{e^{v_c \cdot v_w}}{\sum_{c' \in C} e^{v'_c \cdot v_w}} \tag{3}$$

where v_c and $v_w \in R^d$ are vector representations for context c and word w respectively(Goldberg and Levy, 2014).

One limitation of Word2Vec is that each word can only be represented as one single dot in R^d, and many English words have multiple meanings in use.

The Word2Vec model we trained has 50 dimensions, and in order to compare the performance, we also used a pretrained Word2Vec model from Facebook's FastText (https://fasttext.cc/docs/en/english-vectors.html). It was trained with 16B tokens, has 300 dimensions and contains 1M most frequent words.

3.2 Word2GM

As an attempt to solve the limitation of Word2Vec mentioned above, the word embedding with Gaussian mixtures (word2gm)(Athiwaratkun and Wilson, 2017) tried to represent each word with a Gaussian mixture in high-dimensional space R^d:

$$f_w(\vec{x}) = \sum_{i=1}^{K} p_{w,i}\mathcal{N}(\vec{x}; \vec{\mu}_{w,i}; \vec{\Sigma}_{w,i}) \tag{4}$$

The objective function, derived from Word2Gauss(Vilnis and McCallum, 2014) is to minimize the max-margin ranking:

$$\begin{aligned} L_\theta(w, c, c') = &\max(0, m - \log E_\theta(w, c) \\ &+ \log E_\theta(w, c')) \end{aligned} \tag{5}$$

where the term c' is from negative sampling. Several metrics can be applied to measure the similarity of words:

Maximum cosine similarity

$$d(w_i, w_j) = \max_{p,q=1,\ldots,K} \frac{\langle \mu_{i,p}, \mu_{j,q} \rangle}{\|\mu_{i,p}\| \cdot \|\mu_{j,q}\|} \tag{6}$$

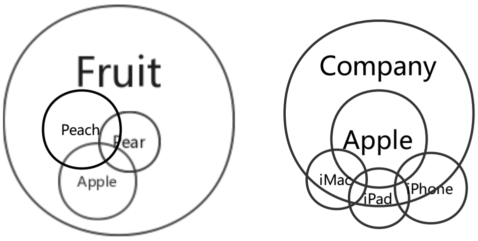

Figure 1: An example of multi-sense words

where p,q are the index of the Gaussian distribution in its Gaussian mixture. Note the cosine similarity is equivalent to the normalized Euclidean distance $\|\cdot\|_2$. We use this metric to evaluate the models.

Expected likelihood(Jebara et al., 2004):

$$\begin{aligned} &E(f(w_i), f(w_j)) = \\ &\sum_{p=1}^{K}\sum_{q=1}^{K} p_{i,p}p_{j,q} \int \mathcal{N}(x;\mu_{i,p};\Sigma_{i,p})\mathcal{N}(x;\mu_{j,q};\Sigma_{j,q})\mathrm{d}x \\ &= \sum_{p=1}^{K}\sum_{q=1}^{K} p_{i,p}q_{j,q}\mathcal{N}(0;\mu_{i,p}-\mu_{j,q};\Sigma_{i,p}+\Sigma_{j,q}) \end{aligned} \tag{7}$$

KL-divergence:

$$\begin{aligned} &D_{KL}(f(w_{i,p})\|f(w_{j,q})) = \\ &\quad \int \mathcal{N}(x;\mu_{j,q};\Sigma_{j,q}) \log \frac{\mathcal{N}(x;\mu_{i,p};\Sigma_{i,p})}{\mathcal{N}(x;\mu_{j,q};\Sigma_{j,q})}\mathrm{d}x \\ &= \frac{1}{2}(d + \log\frac{\det(\Sigma_{i,p})}{\det(\Sigma_{j,q})} - \mathrm{tr}(\Sigma_j^{-1}\Sigma_i \\ &\quad -(\mu_{j,q}-\mu_{i,p})\Sigma_{i,p}^{-1}(\mu_{j,q}-\mu_{i,p})) \end{aligned} \tag{8}$$

Note that KL-divergence is an asymmetric measure of similarity between two distributions.

3.3 Training

We used similar hyper-parameters in Word2Vec and Word2GM training, in order to make the results comparable with each other. Both models applied skip-gram context window $\ell = 5$, space dimensions $D = 50$ and only contained words with word frequency ≥ 5. Training a Word2Vec model is fairly straightforward.

It is yet unclear how to properly choose the number of Gaussians per word K, the boundaries of $\|\mu\|_2$ and Σ and their initial values, thus we simply borrowed the model from the original paper of Word2GM(Athiwaratkun and Wilson, 2017). During training, we observed that the vector space is dense in terms of the radius $\max_i(\|\mu_i\|)$ and covariance matrices Σ, and it might affect the convergence speed of the adaptive gradient descent algorithm we used.

3.4 Testing

We examined the task performance on several models. Firstly, we trained a Word2Vec model with 250M English Wikipedia corpus. Secondly, we trained a Word2GM model with the same Wikipedia corpus and a Word2GM model with a trimmed 100M English Wikipedia corpus since training Word2GM models is quite time-consuming and memory-expensive, and we're currently investigating how to optimize the Gaussian mixture embedding algorithm. Finally, we compared these models with a pre-trained Word2Vec model from Facebook FastText.

4 Evaluation

Model	Accuracy	Score
w2v-250m	0.64	**1.55**
w2v-fb	**0.68**	1.66
w2gm-250m	0.54	2.09
w2gm-100m	0.50	2.25

Table 1: Task 1 Evaluation

As shown in Table 1 and Table 2 the Word2Vec models generally out-perform Word2GM models. The accuracy was calculated based on the best 1 prediction per OOV word while the score takes all the predictions into consideration. In Task 2, only the 5 best predictions for each OOV word were scored since making more prediction attempts would drop the scores. The result of Task 2 indicates that the unsupervised model outputs are still far from how humans use English. Therefore,

Model	Score
w2v-250m	**4.8%**
w2v-fb	3.0%
w2gm-250m	3.2%
w2gm-100m	2.2%

Table 2: Task 2 Evaluation

we want these results to become the baseline of the tasks.

5 Related Work

(Ciaramita and Johnson, 2003) proposed supersense to classify common nouns that extends named entity classification. Based on supersense, (Curran, 2005) improved the rule-based model by adding hand-coded unseen nouns tagging within the scope of WordNet. (Biran et al., 2011) built the text simplification system closely related with WordNet to obtain the word pairs as hypernym or synonym.(Specia et al., 2012) compiled the corpus based on the lexical substitution at SemEval-2007 with manually annotation. However, OOV prediction task is similar with lexical simplification whilst the difference is obvious. First, it is required to acquire prior knowledge as much as better for text simplification. Few OOV words in our task exist in WordNet, not to mention the creation of the thesaurus. With a never-seen word, the key to understand it is to familiarize with the given context. Second, most of the test words and the candidates in lexical substitution tasks such as(McCarthy and Navigli, 2007)are daily words. Thus, the demand for comprehension related to the whole context is less essential than ours.

OOV prediction obstructs the text representation because learning representation in word is the cornerstone for the further text understanding in human sense. In the past, one-hot encoding(Manning et al., 2008) has been used for the word indexing because of the simplicity. Nevertheless, the limitation of the one-hot encoding is apparent. This algorithm produced a sparse matrix for each word representation, but the computational power is much weak, which means it is impossible for the applications with big data. Besides, this method cannot extract the relationship between words, let alone the sentences. Compared with one-hot encoding, word embedding(Rumelhart, 1986)(Bengio et al., 2003) utilizes semantic and syntactic information, which can extract more relationship than one-hot encoding.

However, word embedding also has been constrained by the poor hardware development in previous decades and the algorithm's high time complexity. Recently, (Mikolov et al., 2013)proposed two models (skip-gram and continuous bag-of-words) named Word2Vec for effective learning representation. In addition, those methods succeed to represent word from sparse vectors into dense vectors, which favored for next training or clustering. (Wang et al., 2017)compared the detailed performance between sparse and dense vectors in short text classification.

Those models mentioned above suffers a common weakness, each word is mapped to a unique vector. In fact, numerous words can represent multiple senses in different conversations. Consequently,(Reisinger and Mooney, 2010) proposed multi-prototype model for the polysemy. After that, other related models(Huang et al., 2012)(Tian et al., 2014)(Vilnis and McCallum, 2014)(Athiwaratkun and Wilson, 2017) have been inspired such as probabilistic model, Gaussian and Gaussian mixture models and so on.

6 Conclusions and Future Work

This is the first attempt to address the issues of OOV lexical prediction in NLP tasks with unstructured data and given no prior knowledge. Hence, we propose two tasks for OOV classification and prediction, then we create baseline results with several models based on Word2Vec and Word2GM algorithms. Our result has shown that, OOV lexical prediction is still challenging with unsupervised word embedding models.

We consider the following future directions:

- Improve the Gaussian mixture embedding models such as the hyperparameter selection, loss function optimization, and integration with neural network architectures.

- Expand the data set for the usage of domain specific NLP models and investigate the potential challenges in specific language domains

- Since the models we currently utilizing cannot perform a satisfying result, exploring a novel model is a worthy consideration.

References

Ben Athiwaratkun and Andrew Gordon Wilson. 2017. Multimodal word distributions. In *Conference of the Association for Computational Linguistics (ACL)*.

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

Or Biran, Samuel Brody, and Noémie Elhadad. 2011. Putting it simply: a context-aware approach to lexical simplification. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2*. Association for Computational Linguistics, pages 496–501.

Massimiliano Ciaramita and Mark Johnson. 2003. Supersense tagging of unknown nouns in wordnet. In *Proceedings of the 2003 conference on Empirical methods in natural language processing*. Association for Computational Linguistics, pages 168–175.

James R Curran. 2005. Supersense tagging of unknown nouns using semantic similarity. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*. Association for Computational Linguistics, pages 26–33.

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2006. The pascal recognising textual entailment challenge. In *Machine learning challenges. evaluating predictive uncertainty, visual object classification, and recognising tectual entailment*, Springer, pages 177–190.

Yoav Goldberg and Omer Levy. 2014. word2vec explained: deriving mikolov et al.'s negative-sampling word-embedding method. *arXiv preprint arXiv:1402.3722* .

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*. Association for Computational Linguistics, pages 873–882.

Tony Jebara, Risi Kondor, and Andrew Howard. 2004. Probability product kernels. *Journal of Machine Learning Research* 5(Jul):819–844.

Christopher D Manning, Prabhakar Raghavan, and Hinrich Schütze. 2008. Boolean retrieval. *Introduction to information retrieval* pages 1–18.

Diana McCarthy and Roberto Navigli. 2007. Semeval-2007 task 10: English lexical substitution task. In *Proceedings of the 4th International Workshop on Semantic Evaluations*. Association for Computational Linguistics, pages 48–53.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*. pages 3111–3119.

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*. Association for Computational Linguistics, pages 109–117.

DE Rumelhart. 1986. David e. rumelhart, geoffrey e. hinton, and ronald j. williams. *Nature* 323:533–536.

Advaith Siddharthan. 2006. Syntactic simplification and text cohesion. *Research on Language & Computation* 4(1):77–109.

Lucia Specia, Sujay Kumar Jauhar, and Rada Mihalcea. 2012. Semeval-2012 task 1: English lexical simplification. 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*. Association for Computational Linguistics, pages 347–355.

Fei Tian, Hanjun Dai, Jiang Bian, Bin Gao, Rui Zhang, Enhong Chen, and Tie-Yan Liu. 2014. A probabilistic model for learning multi-prototype word embeddings. In *COLING*. pages 151–160.

Luke Vilnis and Andrew McCallum. 2014. Word representations via gaussian embedding. *arXiv preprint arXiv:1412.6623* .

Ye Wang, Zhi Zhou, Shan Jin, Debin Liu, and Mi Lu. 2017. Comparisons and selections of features and classifiers for short text classification. In *IOP Conference Series: Materials Science and Engineering*. IOP Publishing, volume 261, page 012018.

Ziming Zhang and Venkatesh Saligrama. 2015. Zero-shot learning via semantic similarity embedding. In *Proceedings of the IEEE International Conference on Computer Vision*. pages 4166–4174.