Evaluation of Domain-specific Word Embeddings using Knowledge Resources

Farhad Nooralahzadeh, Lilja Øvrelid, Jan Tore Lønning
Department of Informatics, University of Oslo
{farhadno,liljao,jtl}@ifi.uio.no

Abstract

In this work we evaluate domain-specific embedding models induced from textual resources in the Oil and Gas domain. We conduct intrinsic and extrinsic evaluations of both general and domain-specific embeddings and we observe that constructing domain-specific word embeddings is worthwhile even with a considerably smaller corpus size. Although the intrinsic evaluation shows low performance in synonymy detection, an in-depth error analysis reveals the ability of these models to discover additional semantic relations such as hyponymy, co-hyponymy and relatedness in the target domain. Extrinsic evaluation of the embedding models is provided by a domain-specific sentence classification task, which we solve using a convolutional neural network. We further adapt embedding enhancement methods to provide vector representations for infrequent and unseen terms. Experiments show that the adapted technique can provide improvements both in intrinsic and extrinsic evaluation.

Keywords: word embeddings, intrinsic and extrinsic evaluation, domain knowledge resource, embeddings enhancement

1. Introduction

Domain-specific, technical vocabulary presents a challenge to NLP applications. Recently, word embedding models have been shown to capture a range of semantic relations relevant to the interpretation of lexical items (Mikolov et al., 2013b) and furthermore provide useful input representations for a range of downstream tasks (Collobert et al., 2011). The majority of work dealing with intrinsic evaluation of word embeddings has focused on general domain embeddings and semantic relations between frequent and generic terms. However, it has been shown that embeddings differ from one domain to another due to lexical and semantic variation (Hamilton et al., 2016; Bollegala et al., 2015). Domain-specific terms are challenging for general domain embeddings since there are few statistical clues in the underlying corpora for these items (Bollegala et al., 2015; Pilehvar and Collier, 2016).

The Oil and Gas domain is a highly technical and data-intensive domain. Experts working within this domain daily investigate selected geographical areas and use relevant information (scientific articles, reports and other textual sources) to evaluate the potential for undiscovered hydrocarbons. The vocabulary is technical and there is a real need for NLP tools to aid the work process. In this work we investigate whether word embedding models can capture domain-specific semantic relations by training domain-specific embeddings[1] and evaluating these against a terminological resource. We conduct a comprehensive study including a wide range of evaluation criteria, contrasting several general and domain specific embedding models. We augment the domain-specific embeddings using a domain knowledge resource. To supply embeddings for rare words, we extend the retrofitting method by Faruqui et al. (2015). We then go on to examine the contribution of these models in the performance of a downstream classification task.

[1]Link to the domain-specific model: `http://vectors.nlpl.eu/repository/11/75.zip`

2. Related work

Despite the pervasive use of word embedding in language technology, there is no agreement in the community on the best ways to evaluate these semantic representations of language[2]. There exist a variety of benchmarks which are widely employed to assess the quality of word representations and to compare different distributional semantic models. Existing evaluation methods can largely be separated into two categories: "intrinsic evaluation" and "extrinsic evaluation". *Intrinsic evaluation* tries to directly quantify how well various kinds of linguistic regularities can be detected with the model independent of its downstream applications (Baroni et al., 2014; Schnabel et al., 2015). On the other hand, the quality of a word vector may be assessed by its performance in downstream tasks through measuring changes in performance metrics specific to the tasks by *extrinsic evaluation*. The downstream language technology tasks on which the quality of a word embedding is examined, fall into syntactic (e.g. POS tagging, Chunking) and semantic (e.g. Named entity recognition, Sentiment Classification) categories (Schnabel et al., 2015; Chiu et al., 2016b). In this work we evaluate domain-specific word embedding models using both intrinsic and extrinsic evaluation schemes.

Although, word embeddings techniques have drawn significant interest in the field, they are not well equipped to deal with unseen and infrequent words, nor do they consider word relations found in knowledge resources. Recently, different solutions have been proposed to overcome these limitations (Pilehvar and Collier, 2016; Faruqui et al., 2015; Yu and Dredze, 2014). Among these, we choose Faruqui et al. (2015) in this work since it is a post-processing approach which is straightforward to apply.

3. Intrinsic evaluation setup

Intrinsic evaluation of word embeddings has two requirements. First, we require a query inventory as a gold standard, and second, a word embedding model that has been

[2]RepEval @ACL 2016: The First Workshop on Evaluating Vector Space Representations for NLP

Source	Abbr.	Description	Docs	Sentences
American Association of Petroleum Geologist	AAPG	Scientific articles	3,382	72,243
C&C Reservoirs-Digital Analogs	CCR	Field evaluation reports	1,140	244,017
Elsevier	ELS	Scientific articles, magazines	40,757	7,703,447
Geological Society, London Memoirs	GSL	Scientific articles	152	32,352
Norwegian Petroleum Directory	NPD	Norwegian Field info	514	49,426
Tellus	TELLUS	Basin info	1,478	179,450
Total			47,423	8,280,935

Table 1: Sources of the Oil and Gas corpus

trained on a specific corpus. In this section we describe how we build a domain specific query inventory by exploiting a domain-specific knowledge resource. Then, the domain specific corpus and the training of the embedding models will be described. We then go on to clarify the evaluation methodology.

3.1. Domain specific query inventory

For the general domain, there exists a wide range of gold standard resources for evaluating distributional semantic models in their ability to capture semantic relations of different types, for instance, *Simlex-999* (Hill et al., 2015). However, evaluating the domain specific embeddings by applying these gold standards will not provide an adequate picture of their quality, since they do not share a common vocabulary and word meanings. For this reason, we create a domain-specific gold standard using the Schlumberger oilfield glossary (*slb*).[3] The *slb* is a reference which defines major oilfield activities and has been created by technical experts. Terms are described by their part of speech, their discipline (e.g. *Well Completions*, *Geology*), as well as a textual definition. Terms are linked to other terms in the glossary by means of semantic and lexical relations such as *Synonyms*, *Antonyms* and *Alternative forms*. It provides a network of related terms that can be navigated through the glossary. We construct a domain query inventory by extracting all terms and their inter-glossary relations from the relational database. The glossary consists of *4,886* terms. Following the symmetric nature of the *Synonym*, *Antonym* and *Alternative form* relations we infer a relation if it is missing between terms. The final query inventory contains *878* synonym pairs, *284* antonym pairs and *934* alternative form pairs. We observe that the majority of terms in the query inventory are multi-word units (70%) and nouns (72%). This indicates that a large portion of the domain-specific vocabulary that we want to capture in our model consists of multi-word entities. Thus we should take this into account during the training of embeddings.

3.2. Corpora and Pre-processing

In order to train domain-specific embeddings we need a domain-specific corpus. We therefore compile a corpus consisting of technical reports and scientific articles in the Oil and Gas domain. Table 1 shows detailed information about these sources. The corpus contains $47,423$ documents and $8,280,935$ sentences. It is pre-processed using the following steps: 1) Tokenization and lemmatization using StanfordCoreNLP (Manning et al., 2014). English stop words and sentences with less than three words are also removed from the corpus. 2) Shuffling: we randomly shuffle the text in the dataset. During the training of embedding models the learning rate is linearly dropped as training progresses, text appearing early has a larger effect on the model. Shuffling makes the effect of all text almost equivalent (Chiu et al., 2016a).

3.3. Training of Word Embeddings

For training of the word embeddings, we exploit the available word2vec (Mikolov et al., 2013a) implementation *gensim* (Řehůřek and Sojka, 2010). The elements that have an impact on the performance of the model are the input corpus, model architecture and the hyper-parameters. In many works lemmatized, lowercased and shuffled input during training the word2vec are recommended; we carried out our experiments with these settings as detailed above. We employed the phrase model of *gensim* which automatically detects common phrases (multi-word expressions). The phrases are collocations (frequently co-occurring tokens) and we consider bi-grams and tri-grams in this extraction process. We further proceed with the domain specific model generation by creating two sets of embeddings, employing both the *CBOW* and the *Skip-gram* architectures with default settings. In the initial evaluation step, we compare the outcomes of these two models to determine the better architecture.

We then go on to compare different settings for the hyper-parameters, while keeping all other settings constant. It has been shown that optimizations of hyper parameters and certain system choices constitute the main causes of differences in performance rather than the algorithms themselves (Levy et al., 2015). Here we investigate the impact of various system design choices in the evaluation of domain specific embeddings across the following parameters [4]: I) Vector size: $dim \in 50, \mathbf{100}, 200, 300, 400, 500, 600$ II) Context window size: $win \in 2, 3, \mathbf{5}, 10, 15, 20$. III) Negative sampling size: $neg \in 3, \mathbf{5}, 10, 15$. IV) Frequency cut off: $min.count \in 2, 3, \mathbf{5}, 10$. V) n-most-similar: The parameter n for top n-most-similar as output is fixed at value 5 (the maximum number of terms that are involved in each relation set in the query inventory). We evaluate these different system design settings based on our intrinsic benchmark. We build various embeddings models by varying values of

[3] http://www.glossary.oilfield.slb.com/

[4] Default values are in bold

Model	Synonymy			Antonymy			Alt. form		
	A	R	P	A	R	P	A	R	P
Skip-gram	9.8	8.0	2.2	46.4	41.3	9.3	12.1	10.4	2.4
CBOW	**12.7**	**10.2**	**2.7**	**55.3**	**49.2**	**11.1**	**12.8**	**11.0**	**2.6**

Table 2: Evaluation results for different architectures

one hyper-parameter and keeping others as default. Thereafter, we perform evaluation over the domain specific query inventory.

3.4. Evaluation

For evaluation, we assume that for each term in the inventory an embedding model should be able to propose similar words which are related semantically as either *synonym*, *alternative form* or *antonym*. We will measure this by looking at a target word's relation set, for instance its synonyms, and top n-most-similar words based on the embeddings model. Since these relations are symmetric, the pairs (t_i, t_j) and (t_j, t_i) are considered equivalent in the evaluation. We calculate the *accuracy* (A) as the number of target words for which the model provides at least one correct prediction, the *recall* (R) as the number of correctly predicted word pairs over all word pairs and *precision* (P) as the number of correctly predicted word pairs over all predicted word pairs for each relation category.

4. Intrinsic evaluation experiments

In the following we present experiments that evaluate the domain-specific word embedding models intrinsically. We first present tuning experiments and then present an experimental comparison between domain-specific and general domain embedding models.

4.1. Model architecture: Skip-gram vs. CBOW

First, we compare the models obtained using different architectures (CBOW and Skip-gram) with default values for hyper-parameters i.e. $dim = 100$, $win = 5$, $min.count = 5$ and $neg = 5$. Table 2 presents the results for the two architectures broken down by semantic relation from the query inventory. In general we find that the CBOW based model shows better results than the Skip-gram in all semantic relation tasks. The results show that the embedding models have higher scores for *antonymy* prediction than *synonymy*, see Table 2. This result is consistent with previous studies such as van der Plas and Tiedemann (2006) and Leeuwenberg et al. (2016) in which they reported that using distributional similarity some word categories like antonyms, (co)hyponyms or hypernyms show up more often than synonyms.

4.2. Hyper-parameter tuning

We explore the impact of each hyper-parameter on detection of semantic relations. We observe that the performance of the embedding models can be notably improved over the default hyper-parameters but like the findings in other studies (Chiu et al., 2016a; Gladkova et al., 2016), the effects of different configurations are diverse and sometimes they are counter-intuitive. For example, different relation categories benefit from different context windows size in different ways, such as the model with larger context windows tends to capture antonymy relation while with smaller windows, learns synonymy relation of the words. On the other hand, negative sampling and frequency cut-off parameters have different impacts in the three relation categories.

dim	Synonymy			Antonymy			Alt. form		
	A	R	P	A	R	P	A	R	P
50	12.7	10.2	2.7	48.2	42.9	9.6	11.4	9.8	2.3
100	12.7	10.2	2.7	55.4	49.2	11.1	12.9	11.0	2.6
200	14.7	12.4	3.3	55.4	49.2	11.1	14.3	12.3	2.9
300	**15.7**	**13.1**	**3.5**	55.4	49.2	11.1	13.6	11.7	2.7
400	**15.7**	**13.1**	**3.5**	**57.1**	**50.8**	**11.4**	13.6	11.7	2.7
500	14.7	12.4	3.3	53.6	47.6	10.7	**15.0**	**12.9**	**3.0**
600	14.7	12.4	3.3	51.8	46.0	10.4	12.9	11.0	2.6
700	14.7	12.4	3.3	53.6	47.6	10.7	13.6	11.7	2.7

Table 3: Evaluation results for different vector size (default=100)

win	Synonymy			Antonymy			Alt. form		
	A	R	P	A	R	P	A	R	P
2	12.7	10.2	2.7	55.4	49.2	11.1	**13.6**	**12.3**	**2.9**
3	**13.7**	**12.4**	**3.3**	48.2	42.9	9.6	11.4	9.8	2.3
5	12.7	10.2	2.7	55.4	49.2	11.1	12.9	11.0	2.6
10	13.1	10.9	2.9	53.6	47.6	10.7	**13.6**	**12.3**	**2.9**
15	12.7	10.2	2.7	**67.1**	**50.8**	**11.4**	12.9	11.0	2.6
20	12.7	10.2	2.7	53.6	47.6	10.7	12.1	10.4	2.4

Table 4: Evaluation results for different context window size (default=5)

4.2.1. Vector size (*dim*)

The effect of vector size on the trained models is quite similar in all tasks (Table 3). It shows a large improvement in all evaluations when the dimensionality is increased. However, the improvement peaks at 400 for the *synonymy* and *antonymy* predictions and 500 for *alternative form*.

4.2.2. Context window size (*win*)

Table 4 depicts the impact of window size per evaluation task. The embedding model can detect well the synonymy relation in low windows size (w=3) while in antonymy and alternative form tasks the model performance fluctuates between lower and higher window sizes.

4.2.3. Negative sampling (*neg*)

Unlike the practical recommendation in Levy et al. (2015) which states that the skip-gram model prefers many negative samples, the CBOW model shows contradictory result with respect to this parameter in our evaluation benchmarks. As we can see in Table 5, results remain constant regardless of negative sampling number in the synonym prediction task. While its performance has correlation with an increase of this parameter in alternative form detection. For the antonym task, it reached a peak on neg equal to 5 and 10 before falling.

4.2.4. Frequency cut off (*min.count*)

The impact of excluding words that are less frequent regarding to the $min.count$ parameter is summarized in Table 6. This parameter shows different impact compared to the other parameters. While, ignoring more words has better effect in synonymy detection, it stops at $min.count = 3$ for antonymy and alternative form relations.

neg	Synonymy			Antonymy			Alt. form		
	A	*R*	*P*	*A*	*R*	*P*	*A*	*R*	*P*
3	12.7	10.2	2.7	53.6	47.6	10.7	12.1	10.4	2.4
5	12.7	10.2	2.7	**55.4**	**49.2**	**11.1**	12.9	11.0	2.6
10	12.7	10.2	2.7	**55.4**	**49.2**	**11.1**	13.0	11.7	2.7
15	12.7	10.2	2.7	51.8	46.0	10.4	**13.6**	**12.3**	**2.9**

Table 5: Evaluation results for different number of negative samples (default=5)

min.count	Synonymy			Antonymy			Alt. form		
	A	*R*	*P*	*A*	*R*	*P*	*A*	*R*	*P*
2	12.4	9.9	2.7	54.4	48.4	10.9	13.0	11.8	2.7
3	12.6	10.1	2.7	**56.1**	**50.0**	**11.2**	**13.2**	**12.0**	**2.8**
5	12.7	10.2	2.7	55.4	49.2	11.1	12.9	11.0	2.6
10	**13.1**	**10.4**	**2.8**	54.7	48.3	10.9	13.0	11.8	2.7

Table 6: Evaluation results for different value for frequency cut off (default=5)

Since the context window size (*win*), negative sampling (*neg*) and frequency cut off (*min.count*) parameters showed inconsistent results among the relations, we selected the CBOW model with vector size (*dim*) equal to 400 and we fixed the other parameters to their defaults i.e. $win = 5$, $min.count = 5$ and $neg = 5$. This configuration, hereinafter referred to as OILGAS.*d400*, showed the maximum improvement during the tuning experiments.

4.3. Comparative evaluation

In order to compare the domain-specific embeddings with general domain embeddings, we select two widely used pre-trained embedding sets: *Wiki+Giga* [5] and *GoogleNews* [6] to see how they perform in our evaluation benchmark. The input data in the *Wiki+Giga* has been tokenized and lowercased with the Stanford tokenizer, whereas the *GoogleNews* model is trained on a part of the Google News dataset and it contains both words and phrases. The phrases are obtained using the same approach as described in Section 3.2.. The words are not lemmatized in both models and the GoogleNews also contains capitalized words.

The results of the comparative evaluation of the domain-specific and pre-trained models are summarized in Table 7. Since the words in the vocabularies of both pre-trained models are not in lemma form, we consider the surface form of terms for the evaluation. We also report the proportion of query terms that are covered by the vocabulary of each model as coverage. We find that in spite of the large input and vocabulary size in both *GoogleNews* and *Wiki+Giga* models, they have less coverage than the domain specific model. We further observe that despite the considerably smaller training data set, the OILGAS.d400 performs better across all the tasks.

It is clear that this comparison is somewhat unfair due to differences in pre-processing and hyperparameter tuning. In order to investigate the impact of these differences, we apply the same pre-processing steps and hyperparameters to train the CBOW model over the English Wikipedia dump (20 September 2016), here dubbed *enwiki*. Furthermore, we conduct a similar experiment with a data set consisting of both the general and domain specific corpora (*enwiki+*OILGAS). However, these approaches do not show further improvements in our evaluation benchmark, as reported in Table 7. Surprisingly, the mixing of Wikipedia and OILGAS does not increase the coverage rate. It can be attributed to the fact that the phrase extraction method (Section 3.3.) is not able to capture the multi-word expressions, since in many case in mixed corpus the relative increase in the frequency of tokens individually is higher than relative increase of co-occurring tokens (e.g. the relative increase of the word "source" and the word "rock" in the *enwiki+*OILGAS are bigger than relative increase of the word "source rock" compared to the OILGAS corpus)

[5] https://nlp.stanford.edu/projects/glove/

[6] https://code.google.com/archive/p/word2vec/

5. Error Analysis

The results in Section 4.3. show that the domain-specific model provides better results than general domain models for a domain-specific benchmark. However, we also observe that performance is low for all three tasks, in particular for the synonymy detection task. In this section, we explore the reasons behind these low scores and gain insight into the domain specific model predictions, in particular the synonymy detection, through an in-depth error analysis.

As noted above, the primary cause of low performance is due to out of vocabulary (OOV) terms in the query inventory. The model vocabulary contains only 31% of the evaluation dataset. We find that the majority of terms that participate in synonymy relations are not included in the word embeddings model, this is in particular the case for multi-word items. The majority of these terms either do not occur or have a frequency lower than the cut off threshold in the domain dataset. Excluding the OOV terms from the evaluation tasks has an impact on the model performance for synonymy detection, recall (R) is 29% and precision (P) is 6.5%. Still these scores are low, we therefore examine the model predictions closer.

We choose randomly 100 terms from the reference inventory which are also in the model vocabulary and we manually categorize their 10-most-similar words provided in the word embeddings. In this section, we are inspired by the work of Leeuwenberg et al. (2016), where the authors categorized the result of embeddings for a synonym extraction task in the following categories (The categories with $*$ are added by us).

Spelling Variant: The prediction is an abbreviation or there are differences between prediction and target word because of hyphenation.

Alternative or derived form: The prediction is an alternative or derived form of the target word.

Reference-Synonyms: The prediction is a synonym of the target word in the oilfield glossary.

Human-judged Synonyms: The prediction is judged as true by the expert (but is not present in the glossary).

$*$**Antonyms**: The prediction is an antonym of a target term.

Hypernyms: The prediction is a more general category of the target term.

Hyponyms: The prediction is a more specific type of the target term.

Co-Hyponyms: The prediction and target term share a common hypernym.

Model	Coverage	dim	Synonymy			Antonymy			Alt. form		
			A	R	P	A	R	P	A	R	P
Google News	26% (100B, 3M)	300	9.0	7.0	1.8	51.2	37.0	8.1	4.1	1.6	0.4
Wiki+Giga	23% (6B, 400K)	300	4.0	3.2	0.8	40.4	43.8	10.2	1.8	3.7	0.8
OILGAS.d400	31% (108M, 330K)	400	**15.7**	**13.1**	**3.5**	**57.1**	**50.8**	**11.4**	**13.6**	**11.7**	**2.7**
enwiki	29% (1.8B, 2M)	400	8.2	6.7	1.8	39.1	33.3	7.5	8.3	8.1	1.9
enwiki+OILGAS	31% (1.9B, 2.3M)	400	11.1	7.8	2.1	55.3	47.7	10.7	8.6	8.9	2.0

Table 7: Results from the intrinsic comparative evaluation of general domain and domain-specific embedding models.

Category	Example [target→ prediction]	1^{st}:10^{th}(%)
1. Spelling Variant	borehole → bore-hole	2.4
2. Alternative or derived form	acidizing→ acidization	3.2
3. Reference-Synonyms	filter cake → mud cake	2.8
4. Human-judged Synonyms	seismometer → seismograph	8.4
5. Antonyms	transgressive → regressive	0.9
6. Hypernyms	acidizing → stimulation	1.3
7. Hyponyms	EOR → In-situ combustion	9.3
8. Co-Hyponyms	EOR → MEOR	13.1
9. Holonyms	shoe→ wellbore	1.1
10. Meronyms	rig → wellhead	2.8
11. Related	Kirchhoff migration → NMO correlation	35.2
12. Unrelated/Unknown	backflow → sediment-laden	19.5

Table 8: Manual error analysis results for the 10-most-similar words

∗**Holonyms** The prediction denotes a whole whose part is denoted by the target term.
∗**Meronyms**: The prediction is a part of the target term.
Related: The prediction is semantically related to target.
Unrelated/Unknown: The prediction and target terms are semantically unrelated.
Table 8 shows the result of this analysis. In general, the result of this analysis shows that the model predictions are semantically meaningful in a majority of cases and all categories except the *Unrelated/Unknown* represent one type of morphosyntactic or semantic relation between terms. Less than 20% of errors are assigned to the *Unrelated/Unknown* category. It reveals that if we consider the count of *human-judged synonyms* as true positives, the actual scores for precision and recall will be considerable higher than the ones that are reported in the evaluation section. Moreover, the embeddings model proposes more synonyms that are not in the reference, even though the reference is provided by manual procedure. The most frequent error type falls in the *related* category. The *hyponym* and *co-hyponym* relations are another frequent error type that were also reported in previous studies (van der Plas and Tiedemann, 2006; Leeuwenberg et al., 2016). The morphosyntactic type of relations such as *Alternative forms*, *spelling variant* cover another type of errors. The error analysis further reveals several meaningful relation types such as *Hypernyms*, *Meronyms* and *Holonyms* that are useful in many downstream applications.

6. Embedding Enrichment Using a Knowledge Resource

Even though the word embeddings clearly capture important semantic relations in the domain, the first experiment shows that the domain technical vocabulary has many elements which are generally disregarded by the distributional representation techniques. Since these approaches rely only on the statistics derived from textual input, they are incapable of providing representations for words which are not seen frequently in the training process. Furthermore, they do not include the valuable information that is accommodated in domain knowledge resources such as semantic lexicons and glossaries. In this section, we address these issues by applying the work of Faruqui et al. (2015) to exploit prior domain knowledge in enhancing the embeddings model, and induce representations for OOV terms. We then go on to evaluate the impact of the refinement method over an unseen terminological resource.

6.1. Embeddings for infrequent terms

Faruqui et al. (2015) proposed the retrofitting method as a post-processing step to apply to existing pre-trained embeddings. The goal is to refine word vector representations to capture relatedness suggested by semantic lexicons while preserving their similarity to the corresponding embeddings. The objective of the retrofitting method is to minimize the following:

$$\Psi(Q) = \sum_{i=1}^{n} \Big[\alpha_i \|q_i - \hat{q}_i\|^2 + \sum_{(i,j)\in E} \beta_{ij} \|q_i - q_j\|^2 \Big]$$

where $\hat{q} \in \hat{Q}$ is the observed vector representation for each term in the semantic lexicon and $q \in Q$ is the corresponding retrofitted vector. E is the set of relations among the terms in the semantic lexicons. α and β correspond to the relative weights of relation type. Since Ψ is convex in Q, an

Model	Synonymy			Narrower			Broader			Abbr. label		
	A	R	P	A	R	P	A	R	P	A	R	P
OILGAS	25.5	15.5	5.1	12.5	5.0	2.7	4.7	4.4	0.9	2.4	2.3	0.5
OILGAS.retrofitted	27.4	16.7	5.5	12.5	5.0	2.7	4.7	4.4	0.9	2.2	2.2	0.4
OILGAS.retrofitted+OOV	**30.2**	**18.4**	**6.1**	12.5	5.0	2.7	4.7	4.4	0.9	2.4	2.3	0.5

Table 9: Evaluation over the GeoSci knowledge resource.

efficient iterative updating method is used to solve the objective function. Retrofitted embeddings Q are initialized to be equal to the observed ones $\hat{Q}$. Then by taking first derivative of Ψ with respect to q_i the following online update is used for 10 iterations in order to reach convergence:

$$q_i = \frac{\sum_{j:(i,j)\in E} \beta_{ij} q_j + \alpha_i \hat{q}_i}{\sum_{j:(i,j)\in E} \beta_{ij} + \alpha_i} \quad (1)$$

The formula computes a new embedding for a term i which is in the pre-trained model and has relations of interest in the semantic lexicon, whereas its neighbours should be part of the pre-trained model. To provide an embedding for OOV words we extend $\hat{Q}$ in each iteration by adding the terms that are in semantic lexicon and connect to the terms that are in $\hat{Q}$ via relations of interest. Since there is no initial vector for these type of words in the observed model, α is set to zero and the online update formula for the OOV terms will be as follows:

$$q_i = \frac{\sum_{j:(i,j)\in E} \beta_{ij} q_j}{\sum_{j:(i,j)\in E} \beta_{ij}} \quad (2)$$

6.2. Test Data

We use another domain related glossary to perform a quantitative comparison of domain-specific word embeddings before and after the retrofitting process. We create a test query inventory using the same approach as explained in Section 3.1. using the Geoscience Vocabularies data set [7], here dubbed *GeoSci*. GeoSci covers the domain of geology and describes geological features, geological time, mineral occurrences, and mining-related features. It relates terms with syntactically and semantically aligned relations such as *Abbreviated label*, *Synonym* [8], *Broader* and *Narrower*. We construct a test query inventory by extracting all terms and their inter-glossary relations from the RDF files. The test set consists of *1,753* terms. It contains *196* synonym pairs, *1,639* broader pairs, *1,584* narrower pairs and *965* abbreviated label pairs. Like the *slb* glossary, the majority of terms are multi-word units (63%).

6.3. Evaluation

We use the structure of the *slb* glossary as prior domain knowledge to enrich the OILGAS.d400 embeddings model that is evaluated as a best candidate in Section 4.. Experiments in Faruqui et al. (2015) showed that including all semantic relations in the retrofitting process has a better impact than having only one of them. We therefore consider connections of a word to its synonyms, alternative forms and antonyms. Moreover, similar to the origin, all α_i are set to 1 and β_{ij} to be $degree(i)^{-1}$.

The Eq.1 is used to retrofit the OILGAS.d400 model by employing the structure of the semantic lexicon (".retrofitted"). To induce word vectors for OOV terms , we carry out the retrofitting process with Eq.2 (".retrofitted+OOV"). Table 9 shows the performance of the model in the test dataset as well as the retrofitted models with two different configurations. We observe that the retrofitting process provides improvement in the *synonymy* relation. The improvement is highest when we consider the adapted version (retrofitted+OOV). Interestingly the retrofitted models have no impact in the *narrower* and *broader* relationships, this can be attributed to the fact that the employed semantic lexicons do not include these kinds of associations to lead the retrofitting process. In the *abbreviated label* relation, there is a slightly negative effect when we apply the original retrofitting process.

7. Extrinsic evaluation

While the intrinsic evaluations attempt to interpret the encoding content of an embedding model in terms of lexical semantic relations, *extrinsic* evaluation investigates the contribution of an embedding model to the performance of a specific downstream task. In this section, we investigate the influence of our domain-specific model in a domain related classification task.

7.1. Classification Data Set

The task of the exploration department in Oil and Gas industry is to find exploitable deposits of hydrocarbons (oil or gas). Geoscientists in the exploration department model the subsurface geography by classifying rock layers according to multiple stratigraphic hierarchies using information from a wide range of different sources. The quality of the analysis depends on the availability and the ease of access to the relevant data. Previous technical studies, reports and surveys are crucial resources in this process.

We collect sentences from exploration textual documents which are then manually labeled with various geological type properties by domain experts. Example 1 shows an example sentence from the data set along with its assigned set of properties.

Example 1. *Submarine fans and deltaic/estuarine facies of the San Juan Formation were deposited during the Maastrichtian regression, which gave way during the Paleocene-Eocene to black marine shales and carbonates of the Vidoño Formation and the shelfal and pro-delta shales of the Caratas Formation.*

Properties: Lithology_RockType, Lithology_Main, DepEnv_Sub, DepEnv_General

[7] http://resource.geosciml.org/

[8] GeoSic vocabulary specifies this relation as *Alternative label*.

Model	DepEnv_Sub	Lith_Main	BasinType	DepEnv_Gen	Facies	DepEnv_Main	Lith_RockType
	$F1$	$F1$	$F1$	$F1$	$F1$	$F1$	$F1$
CNN.rand	28.9	90.6	0.0	0.0	63.0	57.1	68.2
CNN.domain	51.1	91.4	23.9	11.3	71.1	66.3	65.8
CNN.multi.rand	38.0	91.4	7.3	5.0	63.9	58.6	69.9
CNN.multi.enwiki	43.9	90.5	11.3	0.0	61.1	61.4	57.8
CNN.multi.domain	56.2	92.2	**33.8**	**15.0**	71.7	69.4	**72.5**
CNN.multi.retrofitted+OOV	64.0	91.3	11.3	0.0	67.6	68.8	72.2
CNN.multi.domain &retrofitted+OOV	**68.2**	**92.8**	32.0	9.4	**73.4**	**73.5**	71.1
CNN.multi.retrofitted+OOV&domain	53.4	92.6	20.9	10.0	71.8	67.0	70.7

Table 10: Results of the classification task with various configurations

Property	# Sentences
Lithology_ Main	1,193
Lithology_ RockType	191
DepEnv_ General	38
DepEnv_ Main	483
DepEnv_ Sub	298
Facies	387
BasinType	49

Table 11: Classification data set

The resulting data set contains 1,348 sentences in which experts assigned each sentence to 7 different properties. The sentences are pre-processed using the same approach as described in Section 3.2. Table 11 depicts the properties and number of sentences for each. It can be seen that the data set is unbalanced regarding to the properties and that the downstream task is a multi-label classification task.

7.2. Multi-label Classification Model

We use a slight variant of the *Convolutional Neural Network* (CNN) architecture that is proposed by Kim (2014) for a sentence classification task. We keep the value of hyperparameters equal to the ones that are reported in the original work, however we update the dimension of the embeddings layer according to the dimension of the domain-specific embeddings model. Furthermore, since the architecture aims to assign a single label to each sentence, we update the activation function to *sigmoid* at the output layer, which produces a probability for each of the potential properties. During training, these probabilities are used to compute the error, while during testing, we round each of the probabilities to 0 or 1 depending upon a set threshold (0.5).

7.3. Extrinsic evaluation experiments

Like Kim (2014), we run experiments with several variants of the model as follows: **CNN.rand**: As a baseline model, where all words in the embedding layer are randomly initialized and updated in the training process. **CNN.domain**: the embedding layer is initialized with a domain-specific model and fine-tuned for the target task. **CNN.multi.rand**: There are two embedding layers as a 'channel' in the CNN architecture. Both channels are initialized randomly and only one of them is updated during training while the other remains static. **CNN.multi.domain**: Same as before, but the channels are initialized with domain-specific vectors. **CNN.multi.enwiki:** The channels consider the general domain word vectors from section 4.3. using the English Wikipedia data. To deal with effects of an unbalanced dataset and guarantee that each fold in 5-fold cross validation will have the proportion of same classes during training and test, we apply the stratification of multi-label data proposed by Sechidis et al. (2011).

Results of the classification task with various CNN configurations are presented in the first section of Table 10. In general, the multi-channel mode performs better than the single channel setting. The results suggest that having a significant amount of sentences per property assists the CNN model to classify better. The baseline model does not perform well on its own. The use of the pre-trained embeddings model helps the model in property assignment. Particularly, domain-specific embeddings provide higher performance gain in the task-at-hand when it is used in both channels.

We also investigate the influence of the refined word embeddings model in our classification task. **CNN.multi.retrofitted+OOV:** We used the retrofitted domain embeddings including the OOV vectors generation for two channels. One channel is static and the other is non-static. **CNN.multi.domain&retrofitted+OOV:** First channel is initialized with original domain-specific embeddings with static mode and the second makes use of the retrofitted embeddings with a non-static mode. **CNN.multi.retrofitted+OOV&domain:** Same as previous setting, but the channels swap their input. In these experiments, because of having many multi-words as OOV terms in the model, we replaced tokens in the sentences with their bi-gram and tri-gram forms if their combination occurs in the model vocabulary (e.g. 'fracture porosity' is replaced to 'fracture_porosity' as a input unit). The experiment (second section of Table 10) shows that the enhanced embedding models provides better input representations for classes with a sufficient number of instances.

8. Conclusion

In the present work we demonstrate that constructing domain-specific word embeddings is beneficial even with limited input data. Nevertheless, the empirical evaluation shows that the distributional models have low performance in domain-specific synonymy detection, an in-depth manual error analysis reveals the striking ability of the embedding models to discover other semantic relations such as (co)hyponymy, hypernymy and relatedness. We further showed the importance of dealing with rare words in an embedding model in both intrinsic and extrinsic evaluation.

9. Bibliographical References

Baroni, M., Dinu, G., and Kruszewski, G. (2014). 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*, pages 238–247. Association for Computational Linguistics.

Bollegala, D., Maehara, T., and ichi Kawarabayashi, K. (2015). Learning word representations from relational graphs. In *Proceedings of 53rd Annual Meeting of the Association for Computational Linguistics, and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing*, pages 730–740. Association for Computational Linguistics.

Chiu, B., Crichton, G. K. O., Korhonen, A., and Pyysalo, S. (2016a). How to train good word embeddings for biomedical NLP. In *Proceedings of the 15th Workshop on Biomedical Natural Language Processing,BioNLP@ACL*, pages 166–174. Association for Computational Linguistics.

Chiu, B., Korhonen, A., and Pyysalo, S. (2016b). Intrinsic evaluation of word vectors fails to predict extrinsic performance. In *Proceedings of the 1st Workshop on Evaluating Vector-Space Representations for NLP, RepEval@ACL*, pages 1–6. Association for Computational Linguistics.

Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., and Kuksa, P. P. (2011). Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12:2493–2537.

Faruqui, M., Dodge, J., Jauhar, S. K., Dyer, C., Hovy, E. H., and Smith, N. A. (2015). Retrofitting word vectors to semantic lexicons. In *The 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1606–1615. Association for Computational Linguistics.

Gladkova, A., Drozd, A., and Matsuoka, S. (2016). Analogy-based detection of morphological and semantic relations with word embeddings: what works and what doesn't. In *Proceedings of the Student Research Workshop, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 8–15. Association for Computational Linguistics.

Hamilton, W. L., Clark, K., Leskovec, J., and Jurafsky, D. (2016). Inducing domain-specific sentiment lexicons from unlabeled corpora. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 595–605. Association for Computational Linguistics.

Hill, F., Reichart, R., and Korhonen, A. (2015). Simlex-999: Evaluating semantic models with (genuine) similarity estimation. *Computational Linguistics*, 41(4):665–695.

Kim, Y. (2014). Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 1746–1751. Association for Computational Linguistics.

Leeuwenberg, A., Vela, M., Dehdari, J., and van Genabith, J. (2016). A minimally supervised approach for synonym extraction with word embeddings. *The Prague Bulletin of Mathematical Linguistics*, 105:111–142.

Levy, O., Goldberg, Y., and Dagan, I. (2015). Improving distributional similarity with lessons learned from word embeddings. *Transactions of the Association for Computational Linguistic*, 3:211–225.

Manning, C. D., Surdeanu, M., Bauer, J., Finkel, J. R., Bethard, S., and McClosky, D. (2014). The stanford corenlp natural language processing toolkit. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, pages 55–60. Association for Computer Linguistics.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013a). Efficient estimation of word representations in vector space. *Computing Research Repository*, abs/1301.3781.

Mikolov, T., Yih, W., and Zweig, G. (2013b). Linguistic regularities in continuous space word representations. In *Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics*, pages 746–751. Association for Computational Linguistics.

Pilehvar, M. T. and Collier, N. (2016). Improved semantic representation for domain-specific entities. In *Proceedings of the 15th Workshop on Biomedical Natural Language Processing*, pages 12–16. Association for Computational Linguistics.

Řehůřek, R. and Sojka, P. (2010). Software framework for topic modelling with large corpora. In *Proceedings of LREC 2010 workshop New Challenges for NLP Frameworks*, pages 46–50. European Language Resources Association.

Schnabel, T., Labutov, I., Mimno, D. M., and Joachims, T. (2015). Evaluation methods for unsupervised word embeddings. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 298–307. Association for Computational Linguistics.

Sechidis, K., Tsoumakas, G., and Vlahavas, I. (2011). On the stratification of multi-label data. In *Proceedings of the 2011 European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 145–158. Springer.

van der Plas, L. and Tiedemann, J. (2006). Finding synonyms using automatic word alignment and measures of distributional similarity. In *21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 866–873. Association for Computational Linguistics.

Yu, M. and Dredze, M. (2014). Improving lexical embeddings with semantic knowledge. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, pages 545–550. Association for Computer Linguistics.