Biomedical relation inference via embeddings

Alex Derry
Department of Biomedical Informatics
Stanford University
aderry@stanford.edu

Abstract

Understanding the relationships between entities such as drugs, genes, and phenotypes is extremely important in biomedical research in order to derive insights into the mechanisms that drive biological responses. However, our knowledge of these relationships is distributed across millions of unstructured text documents and curated databases are inadequate, making it impossible for researchers to access most of this knowledge. We propose the use of embeddings as a way to map entities and the relations between them into the same high-dimensional space, a novel approach which allows for not only information retrieval, but also a variety of novel analyses such as relationship prediction. Phrases representing relation types were identified using a combination of curated and text-mined phrases, and embeddings were generated using a Word2Vec implementation optimized for the task of relationship modeling. By aggregating the representations of equivalent relations and using the resulting vectors as training data, we were able to accurately identify the relationship between two given entities, simultaneously recovering known results and demonstrating the potential to generate entirely novel hypotheses.

1 Introduction

Understanding how entities such as drugs, genes, and phenotypes (e.g. diseases, side-effects) interact with one another is a crucial problem in the biomedical domain, where relationships are highly complex and knowledge of them primarily exists in the form of unstructured text distributed over millions of research articles across the web. With access to the full breadth of this knowledge, it would be possible to derive systems-level insights into the types of interactions that are possible at a molecular scale as well as the mechanisms that drive higher-level processes such as drug-drug interactions or individual differences in drug response.

The traditional approach for extracting and aggregating relevant relationships from these texts for downstream analysis is through manual curation, whereby a legion of expert curators meticulously read research papers and translate the natural language descriptions into the structured form found in databases such as PharmGKB, Online Mendelian Inheritance in Man (OMIM), and DrugBank [1]. This manual curation process is extremely time-consuming and expensive, and is becoming more so as the rate of scientific literature being published increases—over 800,000 articles have been added to Medline every year since 2015 [2]. Another limitation is the fact that only relationships that curators are specifically looking for can be extracted. These problems motivate the development of an automated approach for the extraction and modeling of relationships from biomedical text that does not rely on any predefined notion of relationship classes, thus enabling *de novo* discovery of relational classes and providing a way to map diverse natural language descriptions of interactions to these classes.

In this work, we propose an embedding-based approach to relationship modeling that mitigates many of the shortcomings of existing methods and provides a platform for novel analysis and discovery. We use Word2Vec to represent all biomedical entities (drugs/chemicals, genes/proteins, and diseases/phenotypes/side effects) as well as the words and phrases that encode the relationships between them as 300-dimensional vectors that exist in a single shared embedding space. The hyperparameters of the Word2Vec model were optimized specifically for biomedical relationship modeling, and the resulting embeddings were used as training data for two supervised learning tasks based on (subject, relation, object) triplets: (1) classify relation given subject and object, and (2) predict object given subject and relation. We show that performance can be improved by averaging the vectors of relations with the same meaning, and demonstrate the ability of this approach to recover known interactions and generate novel hypotheses.

2 Related Work

Many text-mining approaches have been proposed for the extraction of relationships from the biomedical literature. Among the earliest and simplest were based on term co-occurrence [3] or rule-based approaches which scan the text for appearances of pre-specified semantic or syntactic patterns which are then matched to their corresponding relationship type [4]. The pattern-matching approach was improved by Huang et al., who used dynamic programming to find protein-protein interaction (PPI) patterns in large text corpora without the need for hand-crafted rules [5]. Some approaches have applied pattern-matching to the dependency trees generated from sentences to identify relationships [6]. Percha and Altman built a text-mined global network of biomedical relationships (GNBR) based on an algorithm called ensemble biclustering for classification (EBC), a statistical method for grouping relationships (expressed as dependency paths extracted from Medline) into related groups [7, 1]. Each of these groups was assigned a "theme", such as *inhibition* or *activation*, which were shown to correspond well with the relationships found in curated databases. Unlike most other relationship extraction studies, GNBR includes gene-gene, gene-disease, drug-gene, and drug-disease interactions, unlike most other approaches which focus on only one type such as PPI. However, EBC relies on co-occurrence of dependency paths in order to classify relationships, resulting in the successful classification of only 40% of all dependency paths.

Methods based on dependency parses and pattern-matching tend to work well when relationships are expressed very simply (e.g. "Gene A regulates Gene B"), but are not good at extracting long-range relationships or dealing with the complex sentence structures that are frequently used in scientific text. Unlike previous approaches, the unsupervised method described in this paper does not depend on the structural properties of individual occurrences of a particular relationship, and therefore represents entities and the relationships between them based on their overall similarity across the entire corpus. This means that every mention of a particular entity/phrase contributes to its final embedded representation, making it possible to achieve higher recall and reveal relationships that cannot be recovered from dependency parses.

3 Approach

3.1 Phrase extraction

The set of multi-word phrases used for this project were mined from a combination of PubTator [8] (for gene, disease, and drug entities) and the Relation Ontology [9], and then manually augmented based on relationship types observed in the dependency paths found in the GNBR database [10]. We originally used the automated phrase extraction tool from Shang et al. (2017) [11] to expand the phrase set based on this initial positive set, but we observed that this did not impact the performance of our models in training, and if anything resulted in additional noise and complexity. Therefore, we continued with only the database-mined phrases, which consisted of about 5.1 million multi-word phrases representing either entities or relationships.

3.2 Corpus preprocessing

The raw corpus was preprocessed for embedding using the following steps:

1. Split into sentences using the Punkt sentence tokenizer implemented in Python's Natural Language Toolkit (NLTK).
2. Remove remaining punctuation from sentences and convert to lower case.

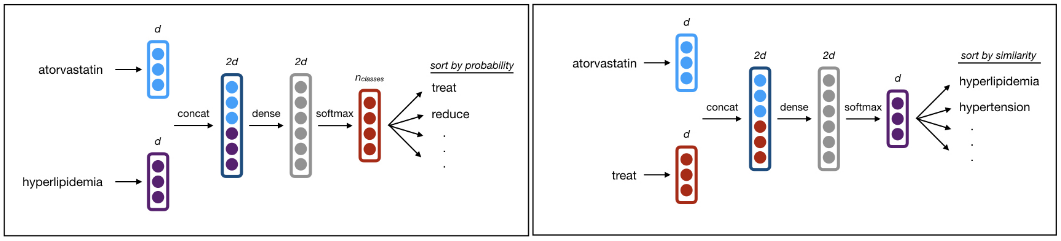
3. Search for matches for list of multi-word phrases (Section 3.1) using the procedure described in the FlashText algorithm [12]. A standard search procedure using regular expressions is $O(m * n)$, where m is the length of the phrase list and n is the length of the corpus. To solve this, all phrases were converted to a trie dictionary before matching, with `<start>` and `<end>` tokens added to each phrase. We can then scan the document one character at a time and traverse the trie dictionary as needed. When a term is completed in the trie dictionary, it is considered a match and returned. All of these matches are then concatenated in the original text, such that *heart disease* becomes *heart_disease*, for example. This procedure reduces the complexity of the phrase matching to $O(n)$.
4. Represent all numbers using a single token, `<num>`. This allows the method to use numbers as a contextual feature, but remain agnostic to the value of the numbers.
5. Stop words were not removed before training; this was an intentional choice made under the hypothesis that some stopwords may be important in the context of relationships. For example, qualifiers such as "might" and "should" could make some relations stronger than others, and negations such as "didn't" or "can't" also have an important meaning.

3.3 Embedding methods

Embeddings were produced using the Word2Vec algorithm, trained using the *gensim* implementation in Python. The input to all runs was the corpus of preprocessed sentences, each tokenized into a list of words. In order to reduce memory load, the corpus was split into 100 files (batches) and read into the model as a generator. The batches were shuffled on each iteration to ensure that the order of sentences in the input did not affect the resulting embeddings. Both skip-gram and continuous bag-of-words architectures were ran for comparison purposes. The best-performing model was selected based on intrinsic metrics (see Section 4) and trained for 40 epochs before extrinsic evaluation.

3.4 Supervised neural relationship prediction

For extrinsic evaluation of the utility of the trained embeddings, we trained two simple feed-forward neural networks for relationship prediction. Given training data consisting of *(subject, relation, object)* triplets, the tasks were to predict the relation given the subject and object entities (Model 1) and to predict the object given the subject and relation (Model 2). Example use cases for each of these models are, respectively, to elucidate the mechanism of action of a drug with a known gene/disease target and to identify the genes that may be inhibited by a particular drug. The input to both models is a concatenated vector of length $2d$, where d is the embedding dimension. The output of Model 1 is a probability distribution (softmax) over all relationship types that appeared in the corpus at least five times ($n_{classes} = 84$, see Section 4.1). This softmax classification architecture is not feasible for Model 2 due to the very large number of possible *object* entities, so the output of Model 2 is a d-dimensional vector with linear activation, which represents the projection of the "predicted word" onto the embedding space. Each model had a single hidden layer with $2d$ neurons, with a ReLU activation function and 50% dropout applied as regularization. The architectures of both models are shown in Fig. 1.



(a) Model 1: Relationship prediction

(b) Model 2: Object prediction

Figure 1: Architectures of neural prediction models, using the example of atorvastatin (Lipitor), a drug for treating high cholesterol.

4 Experiments and Analysis

4.1 Data

The corpus for embedding consisted of ∼28.6 million PubMed abstracts downloaded from PubTator along with their corresponding annotations, which resulted in ∼153.3 million processed sentences. The training data for the supervised prediction models described in Section 3.4 was mined from the dependency paths present in GNBR. Each dependency path in the database corresponds to a pair of entities and represents the connection between the entities in the overall dependency tree of the sentence. Each entity is thus a dependent (either direct or indirect) of the term describing the relationship between them. To generate a high-confidence set of (subject, verb, object) triplets, the relationship term was extracted if it had a direct dependency of type *nsubj* or *dobj*, in which case the entities were assigned to match (Fig. 2. Triplets were also extracted from the Therapeutic Targets Database (TTD), a comprehensive database of drugs and their targets which also includes information about the type of therapeutic relationship (e.g. "inhibitor", "agonist"). Since the relations are derived from natural language, a single relation type is expressed in many different ways depending on grammatical context; for example, the relation "inhibit" appears as "inhibits", "inhibited", and so on. Therefore, to ensure that our prediction networks can treat these as the same fundamental class, we collapse all equivalent words into a single representation of the class. This resulted in 84 classes which were considered legitimate biomedical relations upon inspection.

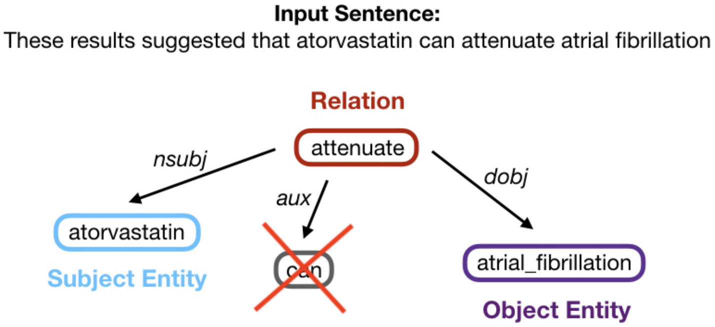


Figure 2: Example of relation extraction, resulting in the triplet (atorvastatin, attenuate, atrial_fibrillation)

4.2 Word2Vec hyperparameter search

The embedding hyperparameters were optimized using a grid search. Based on previous research, the most important hyperparameters to tune for task-specific performance are the window size w (default = 5), the negative sampling coefficient α (default = 0.75), and the down-sampling parameter t (default = 10^{-3}) [13]. Word2Vec has been shown to be quite robust to changes in embedding size, so we held the dimensionality constant at $d = 300$. The parameters tested were $w = 3, 5, 7, 9$, α between -0.75 and 0.75 in intervals of 0.25, and $t = 10^{-5}, 10^{-3}$. The word frequency cutoff was 5. As mentioned previously, each combination of hyperparameters were trained with both skip-gram and CBOW architectures. The resulting 112 models were trained for five epochs and performance was evaluated using intrinsic metrics based on similarity and analogy, which are standard for efficiently estimating the quality of a representation model without the computational complexity required by extrinsic tasks.

4.3 Similarity-based evaluation

The word similarity task is designed to assess the ability of word embeddings to capture basic semantic meaning. Performance is evaluated by measuring the similarities of a list of word pairs with varying degrees of true similarity, and comparing the resulting ranked list with a "gold standard" ranking assigned by humans. Here, we specifically want to assess the similarity between representations of biological words, so we used the *Bio-SimLex* evaluation set, which consists of 250 related and 250 unrelated word pairs from each of the biomedical domain and the general lexicon. The similarity of the 1000 total pairs was annotated by 12 biologists [14]. The ranking produced by each word embedding model was calculated using cosine similarity and compared to the gold standard ranking using Spearman correlation ρ (Fig. 3). These results show that skip-gram tends to perform slightly better ($\sim 3\%$) than CBOW on the similarity task for all values of w and t tested, and has lower variance. This is likely due to its significantly better performance at high and low negative sampling coefficients; at moderate α, CBOW slightly outperforms skip-gram, but performance drops significantly as α decreases. The average Spearman correlations for skip-gram and CBOW with all other parameters set to default were 0.712 and 0.694, respectively, which are very comparable to the benchmarks set in the Bio-SimLex paper (0.715 and 0.698, respectively). The optimal model was a CBOW which achieved $\rho = 0.753$ with $w = 5$, $\alpha = 0.25$, and $t = 10^{-5}$.

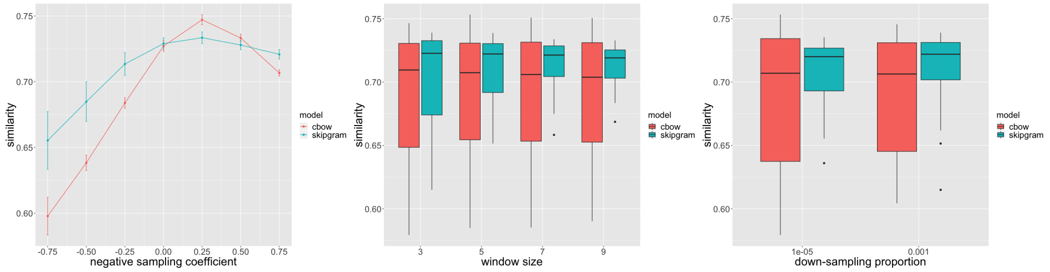


Figure 3: Gridsearch results for Spearman correlation of similarity as a function of negative sampling coefficient, window size, and down-sampling proportion (left–right)

4.4 Analogy-based evaluation

Solving word analogies is a commonly used task to assess the ability of word embeddings to represent relations between entities, which is a closer approximation to the task we are ultimately trying to solve. The typical protocol is to solve the analogy a:b::c:? by finding the closest vector to $c - a + b$. The word corresponding to the predicted vector is given by

$$d_{pred} = argmax_{d \in Vocab}(cos(d, c - a + b)) \tag{1}$$

which is then compared to the true answer to the analogy, and the proportion of correct predictions is the accuracy. The analogy set was created specifically for this analysis. About half of the analogy set was selected from the Biomedical Analogical Similarity Set (BMASS), which was created in a previous study for biomedical analogy evaluation [15]. The rest were generated from relationships extracted from the PharmGKB and TTD, which between them contain gene-gene, drug-gene, gene-disease, and drug-disease relationships. For each of these categories, random subsets were drawn and permuted to produce a set of analogies representing that relationship. The final analogy set consisted of 19 categories with 2450 analogies per category, resulting in a total of 46277 after removing out-of-vocab words.

We consider this simple accuracy metric to be sub-optimal for biomedical analogy evaluation because it is not comprehensive and has high computational complexity due to the need to calculate nearest neighbors on every iteration. However, efforts to improve it have thus far been unsuccessful, so for this paper we used the traditional approach (see Appendix for further details). Results over the grid search are shown in Fig. 4. It is clear that the skip-gram architecture is once again favorable compared to CBOW, performing consistently better over the entire range of hyperparameters. Within architectures, a larger window size resulted in better performance with respect to analogy solving, while the sub-sampling parameter did not have a large effect. Negative sampling coefficient was maximized at $\alpha = 0.5$.

4.5 Assessment of optimal model

Based on the grid search, the optimal model was chosen to be a skip-gram architecture with $w = 9$, $\alpha = 0.25$, and $t = 10^{-5}$. A lower sub-sampling rate was chosen to speed up training, since skip-gram trains significantly slower than CBOW. This model was trained for a total of 25 epochs. As a sanity check to ensure that the final representations are able to capture the key differences between different

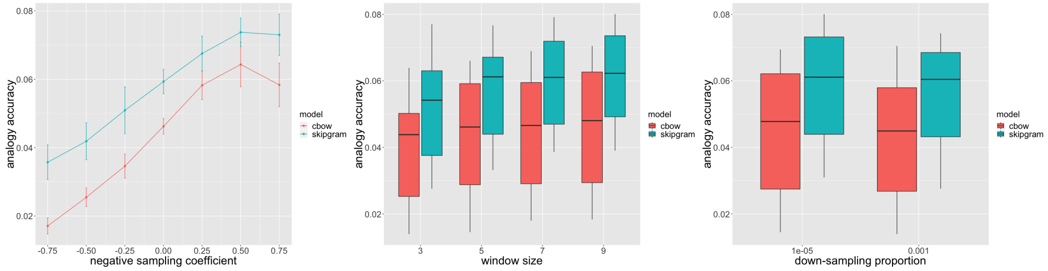
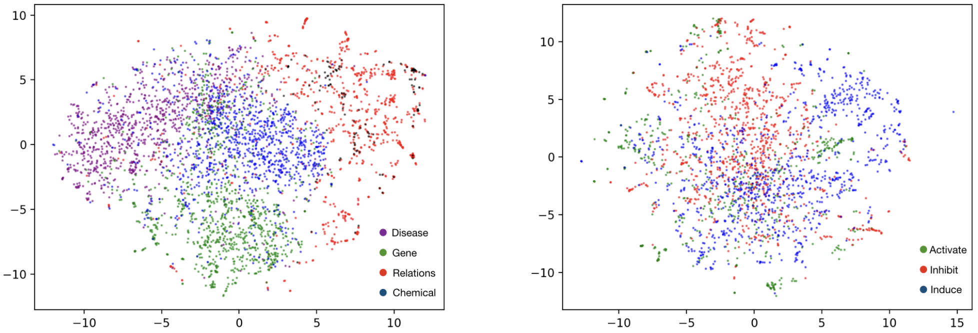


Figure 4: Gridsearch results for analogy accuracy as a function of negative sampling coefficient, window size, and down-sampling proportion (left–right)

types of terms, Fig. 5a shows a 2-D projection of a random sample of word vectors balanced by class for the four key classes of entities: genes, diseases, drugs, and relations. It is clear that these classes are separated in the embedding space. However, within the relations class, there is no clear difference within active tense (e.g. "inhibits"; red) and passive tense (e.g. "inhibited by"; black). This is most likely an artifact of Word2Vec, since these terms will appear in almost identical context windows and so the model cannot distinguish them. This observation motivates the inclusion of both active and passive tenses in the collapsing of words as described in Section 4.1. Ideally we would like to be able to separate these two tenses, because we want to know the direction of interaction between two entities as well as the type; this is therefore an important direction of further research.

Importantly, there does appear to be some separation between different types of relations, as shown in Fig. 5b. The classes are not perfectly separated but they are clearly not randomly distributed, implying that it is possible for a model to learn to distinguish them. This is crucial for the utility of these embeddings, and demonstrates that the skip-gram model is able to differentiate words that are syntactically identical based on their semantic differences. This is probably because we have sufficient training data for the model to learn which entities each relation type tends to co-occur with, and these entities are distinct enough in their features to provide useful information to the model. Both projections were produced by t-distributed stochastic neighbor embedding (tSNE) with two components. To reduce complexity, only the first 50 principle components were used as input to tSNE.



(a) tSNE projections of broad entity classes: genes, diseases, drugs, and relationships

(b) tSNE projections of three common relation types: inhibit, activate, and induce

Figure 5: Embedding space visualizations to demonstrate separability of classes

4.6 Relationship modeling performance

Both models for relationship prediction were trained for 10 epochs with Adam optimizer and a learning rate of 10^{-5} (this was necessary to prevent weight divergence during training). Model 1 (relation classification) was trained with categorical cross-entropy loss because its output is a softmax distribution, while Model 2 (relationship object prediction) was trained with a cosine distance loss function because its output is a projection onto the embedding space and the ultimate objective is to find entities that have high cosine similarity with the predictions. Both models were evaluated before and after collapsing the relation types into classes in order to assess the ability of the model to learn the aggregated representations. The uncollapsed state is considered as a baseline and is expected to result in low performance because the class representations are distributed across the embedding space. Training was performed on 80% of the label data, resulting in ~ 2.7 million examples with no collapsing and ~ 1.2 million examples with collapsed relation labels. The remaining 20% of data was held out for testing.

4.6.1 Model 1 evaluation

Performance on relation classification was evaluated using a strict metric (is the maximum-probability prediction correct?) and a more relaxed metric (does the correct class appear in the top 10 predictions?). The second metric is motivated by the hypothesis-generating aspect of this approach; a scientist who is investigating a pair of entities is interested in a set of predictions about the relationship between them, from which they can potentially discover new relationships or understand a general theme. For example, if all of the top predictions imply a positive correlation between two entities (e.g. "activate", "induce", "upregulate", etc.), that is an important insight that we cannot gain from just a single prediction. In fact, given the noise in our training data, it is likely that a different relation than that considered "truth" here is actually more appropriate in many cases. The training prediction should still appear close to the top however, which is why we present the top 10 accuracy as well as the mean reciprocal rank (MRR). MRR is defined by

$$MRR = \frac{1}{n_{test}} \sum_{i=1}^{n_{test}} \frac{1}{r_i} \tag{2}$$

where r_i is the position of the true label in the ranked list of predicted labels for test sample i. Thus, an MRR of 0.25 means that the correct answer averaged a rank of 4^{th} across the predictions for all samples. All metrics are reported as the mean over 100 random test sets of size $n_{test} = 10000$ drawn without replacement from a much larger test set that was held-out from training.

4.6.2 Model 2 evaluation

The ability to predict the object of a relationship given its subject and the relationship type was assessed using MRR and mean average precision (MAP). Mean average precision is a metric for assessing the precision and recall of the ranked list of predictions produced by a model relative to a set of known true labels. In this case, the model produces a 300-dimensional projection vector, so we generate ranked predictions for all entities using their cosine distance from this vector. The set of true labels for each data point was considered the list of all valid objects from the training data for a given subject and relation. MAP is calculated as

$$MAP = \frac{1}{n_{test}} \sum_{i=1}^{n_{test}} \frac{1}{AP_i} \tag{3}$$

where AP_i is the average of the maximum precision over all recall levels (number of correct answers r):

$$AP = \frac{1}{n_{true}} \sum_{r=1}^{n_{true}} \max_{r* \geq r} prec(r*) \tag{4}$$

In other words, as we move down the ranked list for each test example, each time a correct prediction is encountered, the AP at that recall level is calculated by the number of correct answers so far divided by the number of predictions tested. MRR is defined in the same way as above, using the rank of the first correct answer in the list.

4.6.3 Results

The results for all models tested are shown in Table 1. For relation classification, we are able to predict the correct class label over half the time even without aggregating classes, which is significantly better than would be expected by chance. However, by combining equivalent classes we are able to improve the top-10 accuracy to almost 80% and significantly improve raw accuracy and MRR. The resulting MRR of almost 0.4 implies that the correct answer appears within the top 3 predictions on average. This improvement shows that the model is actually able to learn features of the different relation classes rather than just relying on co-occurrences of specific forms of each class (e.g. present vs. past tense). To see where the model was going wrong, we looked at the relations which most often did not appear in the top 10 predictions. The 5 most frequently missed terms were "modulate", "block", "control", "alter", and "stimulate", with 102, 81, 74, 68, and 65 missed predictions respectively (out of the 10000 total predictions). It is likely that these are frequently missed because they are more general relations or are more commonly used in contexts outside of biomedical interactions, and our model is better at predicting more specific interactions such as inhibition or activation.

For predicting the object of an interaction given its subject and relation, relation aggregation also had a marked effect on performance. However, the overall performance on this task was still lower than that of relationship prediction; while the highest-ranked correct prediction appears just inside the 4^{th} position on average, the average precision for each recall level was less than 5%. While the model may to be able to find one of the correct answers with some precision, it comes at the cost of the others The objective in this case is substantially more difficult because the space of possible entities is in the millions and the projected prediction vector is not specific enough to identify the right target, particularly when there are several right answers. It is likely that the model is placing the prediction near one of the right answers at the expense of being far from the others. To address this, it may be beneficial to group the entities into similar classes and train on the averaged class vectors. These groups could be determined in various ways depending on the desired level of precision; for example, only entities with synonymous meaning or entities that are part of a particular subcategory (e.g. disease subgroups or gene pathways). Additionally, the training data was not processed beyond the initial mining from databases, so increasing quality and reducing noise here would most likely boost performance of both models.

	Model 1: Relation classification			**Model 2**: Object prediction	
	Accuracy (top 1)	Accuracy (top 10)	MRR	MAP	MRR
No class aggregation	0.142	0.561	0.267	2.529×10^{-4}	0.004
With class aggregation	**0.231**	**0.784**	**0.393**	**0.042**	**0.273**

Table 1: Performance of each model before and after combining equivalent relation classes

5 Conclusions

We demonstrated the utility of word embeddings as a way to model relationships between biomedical entities in unstructured text. Using an tuned Word2Vec model with skip-gram architecture, we obtained trained embeddings of genes, drugs, diseases, and the terms that encode the interactions between them, which were then used to train supervised neural networks for relationship prediction. We found that the representations of individual relationship types were sufficient to distinguish between classes of relationships, and that by aggregating equivalent relationships it is possible to significantly boost performance. This gives us confidence that the models are learning actual differences in meaning between interaction classes. For the task of relationship classification, the correct relation was recovered with almost 25% accuracy among 84 possible classes, and was present in the top 10 predictions 78% of the time. Predicting the object of an interaction proved considerably more challenging due to the errors associated with projecting the prediction onto a 300-dimensional space and the very large number of entities to compare the predictions to; improvement on this task will be the subject of future work. Another important direction to pursue is capturing the direction of interactions as well as their nature, which could be achieved by discriminating between active and passive tense. This is probably not possible with Word2Vec, so we would like to explore the possibility of augmenting the embeddings from Word2Vec with those from a context-aware model such as BERT.

With further fine-tuning, the unsupervised approach described here can more effectively model all entities and phrases based on learned characteristics of every sentence in which they exist, thus achieving higher recall than co-occurrence based methods. It also provides the ability to predict novel and/or latent relationships that are not present in existing databases, facilitating the construction of an expanded knowledge graph of biomedical relationships. This hypothesis-generating ability sets our method apart and could be extremely useful for future applications such as drug target discovery, drug repurposing, and side effect prediction for drug-drug interactions.

Acknowledgments

I would like to acknowledge my PI, Dr. Russ Altman, and Adam Lavertu, a PhD student in the Altman lab, for their advice and mentorship in this project. Suvadip Paul was my CS224N grader and mentor.

References

[1] Bethany Percha and Russ B Altman. A global network of biomedical relationships derived from text. *Bioinformatics*, 34(15):2614–2624, 02 2018.

[2] National Library of Medicine (NLM). Detailed indexing statistics: 1965-2017. `https://www.nlm.nih.gov/bsd/index_stats_comp.html`, 2017. Accessed: 2018-03-18.

[3] Blaise TF Alako, Antoine Veldhoven, Sjozef van Baal, Rob Jelier, Stefan Verhoeven, Ton Rullmann, Jan Polman, and Guido Jenster. CoPub Mapper: mining MEDLINE based on search term co-publication. *BMC Bioinformatics*, 6(51), 03 2005.

[4] Akira Tanigami, Haretsugu Hishigaki, Toshihide Ono, and Toshihisa Takagi. Automated extraction of information on protein–protein interactions from the biological literature . *Bioinformatics*, 17(2):155–161, 02 2001.

[5] Minlie Huang, Donald G. Payan, Kunbin Qu, Ming Li, Xiaoyan Zhu, and Yu Hao. Discovering patterns to extract protein–protein interactions from full texts. *Bioinformatics*, 20(18):3604–3612, 07 2004.

[6] Katrin Fundel, Ralf Zimmer, and Robert Küffner. RelEx—Relation extraction using dependency parse trees. *Bioinformatics*, 23(3):365–371, 12 2006.

[7] Bethany Percha and Russ B. Altman. Learning the structure of biomedical relationships from unstructured text. *PLOS Computational Biology*, 11(7), 2015.

[8] Chih-Hsuan Wei, Hung-Yu Kao, and Zhiyong Lu. PubTator: a web-based text mining tool for assisting biocuration. *Nucleic Acids Research*, 41(W1):W518–W522, 05 2013.

[9] OBO Foundry. Relation ontology. `http://www.obofoundry.org/ontology/ro.html`. Accessed: 2018-03-18.

[10] Bethany Percha and Russ B. Altman. A global network of biomedical relationships derived from text, November 2018.

[11] Jingbo Shang, Jialu Liu, Meng Jiang, Xiang Ren, Clare R. Voss, and Jiawei Han. Automated phrase mining from massive text corpora. *CoRR*, abs/1702.04457, 2017.

[12] Vikash Singh. Replace or retrieve keywords in documents at scale. *CoRR*, abs/1711.00046, 2017.

[13] Hugo Caselles-Dupré, Florian Lesaint, and Jimena Royo-Letelier. Word2vec applied to recommendation: Hyperparameters matter. *CoRR*, abs/1804.04212, 2018.

[14] Billy Chiu, Sampo Pyysalo, Ivan Vulic, and Anna Korhonen. Bio-SimVerb and Bio-SimLex: wide-coverage evaluation sets of word similarity in biomedicine. *BMC Bioinformatics*, 19(33), 02 2018.

[15] Denis Newman-Griffis, Albert M. Lai, and Eric Fosler-Lussier. Insights into analogy completion from the biomedical domain. *CoRR*, abs/1706.02241, 2017.

[16] Xiaoyin Che, Nico Ring, Willi Raschkowski, Haojin Yang, and Christoph Meinel. Traversal-free word vector evaluation in analogy space. In *Proceedings of the 2nd Workshop on Evaluating Vector Space Representations for NLP*, pages 11–15. Association for Computational Linguistics, 2017.

Appendix

Improving the analogy evaluation task

There are several problems with the traditional analogy approach. Unlike the analogies typically used in the general domain, which solve relations like *country:capital* or *adjective:noun*, relationships in the biomedical domain are complicated and do not tend to have only one answer. This is particularly true for the relationships we are interested in, such as drug-target relationships. The multi-answer analogy problem has been addressed before in the biomedical domain, including in the BMASS paper. Even so, the multi-answer framework still relies on computing the distance to every word in the vocabulary, which becomes very expensive for a vocabulary with millions of terms. Additionally, these methods all rely on specific matches and thus cannot capture finer-grained differences in relational similarity.

Due to these limitations, we are exploring the use of a different metric for analogy evaluation in the biomedical domain. Since the goal of the analogy task is to determine whether two pairs of entities have a shared relationship type which can be modeled by vector subtraction, instead of calculating $c - a + b$ for each analogy and comparing the result to d, we can directly calculate the similarity between the subtracted vectors $b - a$ and $d - c$ in the vector space [16]. If the relationship between a and b is the same as that between c and d, the subtracted vectors should be very similar (i.e. cosine distance close to 1). The new cosine similarity metric is thus defined as

$$sim = \frac{(b-a)\cdot(d-c)}{|b-a||d-c|} \in [-1, 1] \tag{5}$$

This metric allows for a more nuanced evaluation than accuracy and does not require traversing the entire vocabulary. This reduced the evaluation time for the entire space of 112 grid-search models from $15h57'43$" (~ 485"/model) to $1h22'00$" (~ 45"/model), a significant improvement. However, after calculating the similarity scores, we observed the opposite trend to what was expected based on the traditional accuracy metric. The two metrics should have a positive correlation since they are both estimating the same underlying relationships, but we observe a negative correlation (Fig. 6). The reason for this is currently under investigation, and breaking down the score by analogy category has failed to elucidate the issue.

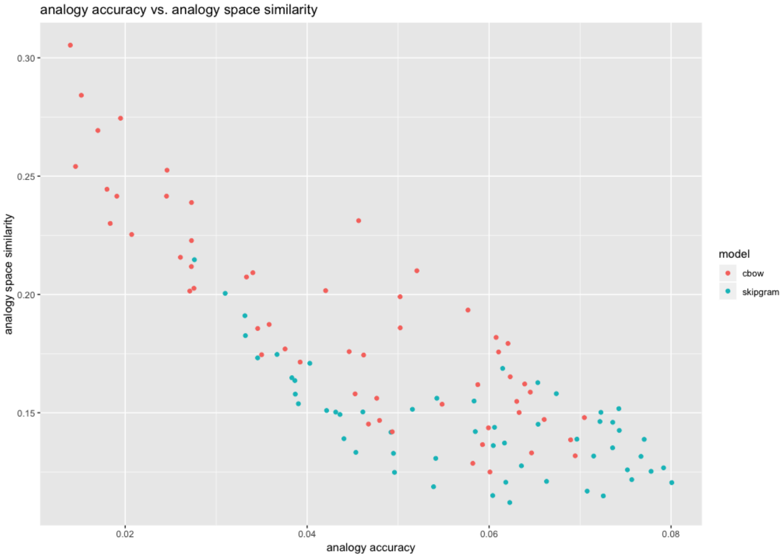


Figure 6: Proposed analogy space metric compared to traditional analogy accuracy