Modelling Semantic Compositionality to make a Semantically Sensible Thesaurus

Kavita Vaishnaw*
IIT Gandhinagar
Gujarat
kavita.vaishnaw@iitgn.ac.in

Shweta Pardeshi*
IIT Gandhinagar
Gujarat
pardeshi.shweta@iitgn.ac.in

Kanishk Kalra*
IIT Gandhinagar
Gujarat
kanishk.kalra@iitgn.ac.in

Rohan Patil*
IIT Gandhinagar
Gujarat
rohan.patil@iitgn.ac.in
*

Mayank Singh[1]
IIT Gandhinagar
Gujarat
mayank.singh@iitgn.ac.in †

Abstract

Semantic Compositionality refers to the principle that meaning of a sentence is determined by the meanings of its constituent words and the rules used to combine them into a sentence. The task of modelling semantic compositionality is to create a mathematical model that is able to understand the relationships in text and natural language like humans. Present methods have used vector based factorization approach to achieve the same but it fails for complex sentences. We propose three approaches that aim to model this task. The first is a tensor based factorization approach proposed by Cruys et al. which is used as our baseline model in which we model the interactions of verbs with subjects and objects using decomposition of subject-verb-object(SVO) co-occurrence tensor. Next we propose a novel classical model based on the idea that whatever SVO triples are observed in the dataset are "Good English", hence the name Good English Model. This model is based on the probability of occurrence of words in the context based on Good English data. An advanced version of this model is also proposed that is able to combine the sense definitions of the words from WordNet with the purely contextual Good English model. In order to analyse the results obtained and compare them using a neural approach, we also implement a pre-trained BERT architecture for this task. All these models are evaluated on a similarity task for transitive phrases and we observe that the Good English Model turns out to be the best on our testing data. These approaches are then used to suggest word-replacements that are "semantically sensible". This is called the Semantically Sensible Thesaurus and provide specialized replacements unlike the regular thesaurus which provides a list of words that simply mean the same.

*Equal Contribution
†Mentor

1 Introduction

According to Stanford Encyclopedia of Philosophy, *"the principle of semantic compositionality refers to the fact that the meaning of a complex expression is fully determined by its structure and the meanings of its constituents"*. While for humans it is trivial to comprehend and make sense of a sentence that they have never come across before, it is a difficult task to automate. The compositionality of text is often modelled using a Bag of Words model ignoring the "compositions" of the sentences in the sense of the actual meaning of words and long term syntactical dependencies. An efficient model to automate this task can be thought of as a sweet marriage between sequence modelling and word-sense disambiguation.

Many vector based distributional models have been proposed for this task which have produced some good results for simple transitive sentences.

However, these models prove ineffective when used with "Natural Language", which is ever-evolving. Modelling this concept could be a potential stepping stone towards automating the creation of semantically correct text. This problem stands at the crossroads of linguistics, computer science, information engineering, and artificial intelligence and aims to contribute to the larger goal of creating machines with the ability to understand and interpret the language as humans.

In this paper, we present a few models for this task which aim to encapsulate the composition of natural language by combining the orthogonal concepts of meaning and context (Mitchell and Steedman, 2015) . The first model is called **Tensor Based Factorization Model of Semantic Compositionality** which is essentially a re-implementation of (Van de Cruys et al., 2013). Here we model the multi-way interactions of verbs with subjects and objects in our sentences using a core tensor obtained from the factorization of a subject-verb-object co-occurrence tensor using latent factors of subjects and objects. The latent factors of nouns(subjects/objects) are calculated using a non-negative matrix factorization of Nouns-Context Words co-occurrence matrix. A latent factor is nothing but a mathematical abstraction representing the relationship between nouns and their context words.

Next we propose a classical model that we call **Good English Model**. The main idea behind this model is that all the sentences that we have seen in our dataset are presumed to be "Good English" as they have been stated by humans. So, this model simply tries to compare any new sentence/text with the Good English that it has already seen and is thus able to model semantic compositionality. This model is completely based on the probability of words with their contexts. We calculate the mutual information of all subject-verb-object triples as well as the subject-verb, verb-object, and subject-object pairs that it has seen and tries the evaluate any new word using this probabilistic information.

Now, we know that most words in the English language do not have only one meaning. Many words have multiple senses that can occur in a sentence. We might be able to capture such different senses using the context if we have enormous amount of training data and resources. If not, we might still work around by using a smoothing technique that takes into account the synonyms, hyponyms and hypernyms of the words using WordNet (Miller, 1995). We call this the **Advanced Good English Model**. In this way, even if we do not have enough data for a particular word, we are able to provide some reinforcement to it by giving it some extra weight according to the overlap in the WordNet synsets. This is also able to account for the inherent bias in our dataset.

In order to compare these models with the current state of the art deep learning models, we implement a pre-trained BERT-based model (Devlin et al., 2019) for the same evaluation technique. Our models are actually comparable to the BERT model in this case as they have been trained on a different dataset. This is just to analyse the results that we observe while working with all these different models.

All our models are evaluated based on a similarity task for transitive phrases. We have a test dataset consisting of test sentences and replacements that have been scored by humans. We calculate the scores for these sentences using our models and then find the Spearman Correlation coefficient between the model scores and the human annotated scores. The evaluation results and analysis is then presented highlighting the insights that we gather from this exercise and the tasks that we can do in the future to efficiently model the semantic compositionality of the natural language.

We introduce a practical use-case for this task that we call, **Semantically Sensible Thesaurus** in which we take a sentence and provide verb-replacements using our models. Looking for synonyms in a thesaurus is not efficient because the same word can have different implications when placed in different sentences. Creating a thesaurus that can suggest synonyms without disturbing the semantics of the text is a difficult task to automate.

2 Related Work

In recent years, a number of methods have been developed that try to capture compositional phenomena within a distributional framework. One of the earliest attempts at models of semantic compositionality was proposed by Mitchell and Lapata in 2008 which is the Vector-based Models of Semantic Composition. They construct additive and multiplicative frameworks for representing the meaning of phrases and sentences in vector space. These models were evaluated on a human anno-

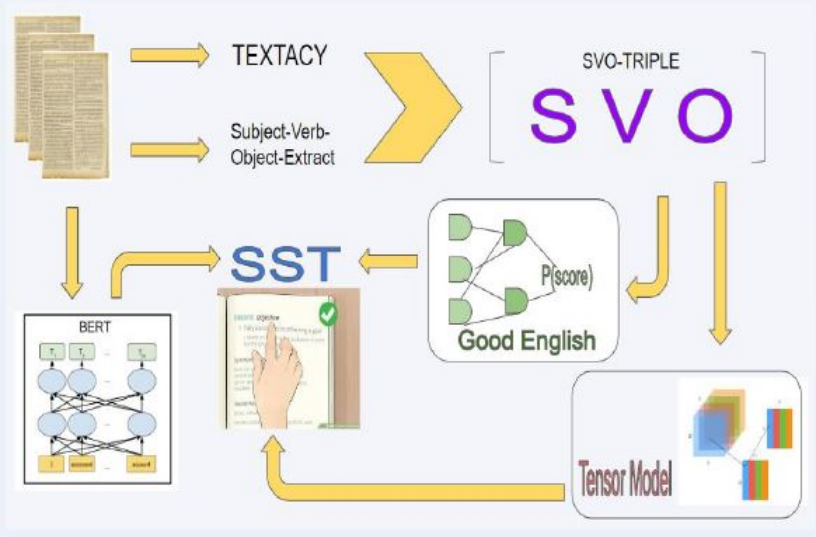

Figure 1: Models

tated sentence similarity task in which the multiplicative model performs better. (Mitchell and Lapata, 2008)

Baroni and Zamparelli (2010) proposed an approach to adjective-noun composition for distributional semantics that represents nouns as vectors and adjectives as matrices. Adjective is considered as a linear transformation from one vector to another vector. (Baroni and Zamparelli, 2010)

Cocke et al. (2010) proposed the mathematical foundations of the compositional distributional model of meaning which enables us to compute the meaning of a well-typed sentence using the meaning of its constituent words. A number of instantiations of this framework were tested experimentally in Grefenstette and Sadrzadeh (2011a) and Grefenstette and Sadrzadeh (2011b). The evalutaion used in our paper is also taken from Grefenstette and Sadrzadeh. (Coecke et al., 2010)

Several neural models have also been proposed for the purpose of this task, the earliest and the most significant being the Socher et al. (2012) which is based on recursive neural networks. Each node in a parse tree is assigned both a vector and a matrix; the vector captures the actual meaning of the constituent, while the matrix models the way it changes the meaning of neighbouring words and phrases. CNN (Kim, 2014) has been used for sentence classification and RNTN (Socher et al., 2012), which is a combination of tensors with RNN, has been used for sentiment analysis, but these methods have not been employed to model the semantic relationships between words. Convolutional Neural Tensor Network (CNTN) has been proposed for encoding sentences in semantic space by Qiu and Huang (Qiu and Huang, 2015) which is also a very novel approach for language processing.

3 Models

Here, we present several models that are able to model the semantic compositionality of text. A summary of all the models is shown in Figure 1. These are explained in detail in the subsequent sections.

3.1 Tensor-Based Factorization Model

Tim Van de Cruys, Thierry Poibeau, and Anna Korhonen (Van de Cruys et al., 2013) modeled semantic compositionality using a tensor-based factorization method. The main idea of their approach is to exploit multi-way interaction between latent factors of subject-verb-object (SVO) triples to model compositionality.

In this model, we first find the latent factors of most frequent nouns and verbs using non-negative matrix decomposition. The latent factors are constructed based on the context words which are within a window of 10 words.

$$V_{nounsXcontexts} = W_{nounsXk} H_{kXcontexts}$$

Where, V is the original matrix and W and H are latent factors.We have used k =300 as given in the

The NMF is done using the iterative method to minimize KL-divergence between the original

matrix and the product of the latent factors.

$$H_{a\mu} = H_{a\mu} \frac{\sum_i W_{ia} \frac{V_{i\mu}}{(WH)_{i\mu}}}{\sum_k W_{ka}}$$

$$W_{ia} = W_{ia} \frac{\sum_\mu H_{a\mu} \frac{V_{i\mu}}{(WH)_{i\mu}}}{\sum_v H_{av}}$$

The next step is to construct SVO triples of the corpus. The SVO tensor is constructed using co-occurrences of lemmatized SVO triples in the corpus. To model multi-way interaction between SVO triples, tucker decomposition is used. Tucker decomposition decomposes a tensor into matrices and a core tensor. Since Tucker Decomposition is computationally expensive, authors have proposed an alternate method to find the core tensor.

$$\mathbb{G} = \chi \times_2 W^T \times_3 W^T$$

Where $\mathbb{G}$ is the core tensor
To evaluate composition of a particular sentence, we first extract SVO triple of the sentence, corresponding subject vector w_s and object vector w_o from the matrix W. After that we carry out the following computations:

$$Y = w_s \otimes w_o$$

Where, Y is the outer product of w_s and w_o.

$$Z = G_v * Y$$

Where, G_v is the corresponding verb slide from tensor $\mathbb{G}$ and $*$ denotes Hadamard product.

We have used 5000 the most frequent nouns and 2000 most fequent context words to construct matrix V. To construct the tensor χ we have used 1000 most freuqent verbs, subjects and objects each . This tensor and matrix are weighted using pointwise Mutual Index (PMI).

$$PMI(s, v, o) = log \frac{P(s, v, o)}{P(s)P(v)P(o)}$$

where:

$$P(x) = \frac{freq(x)}{No.oftokens}$$

$$P(x, y, z) = \frac{freq(s, v, o)}{No.oftokens}$$

3.2 Good English Model

The Good English model is based on classical Natural Language Processing techniques. It assumes that all the sentences that it has seen from the data set are "good", that is, all the subject-verb-object (SVO) triples it has encountered in the data set, are correct.

The model aims to find a verb replacement for the input SVO triple. In order to do that, it substitutes the original verb with the target replacement verb to make a new SVO triple (SV_tO). It then gives a score for how "good" the SV_tO formed is. It takes into consideration the meaning as well as the context of the verb.

We have first processed the data set to extract SVO triples. Simultaneously, we have also extracted the frequencies for SVO, VO, and SV.

The final score given by the model for a new SV_tO triple takes into account the scores for the old SVO, SV, VO, SV_tO, SV_t and V_tO. Firstly, we calculate the probability for these entities. We have included Add-1 smoothing to account for the SVO triples and corresponding pairs that are seen for the first time by the model. The expression for the probability is as follows:

$$P(w) = \frac{freq(w) + 1}{N_w + |V_w|}$$

where
w: can be SVO, SV, VO, SV_tO, SV_t, V_tO
N_w : Total number of w in the dataset
V_w : Total types of w
Afterwards, we have calculated the sigmoid of the PMI (Pointwise Mutual Information) of w and used that as the score. We define the PMI for SVO as it was defined in (Van de Cruys et al., 2013).

$$PMI(S, V, O) = log \frac{P(S, V, O)}{P(S)P(V)P(O)}$$

$$score(w) = \frac{1}{1 + e^{-PMI(w)}} = \frac{MI(w)}{1 + MI(w)}$$

Now, we will calculate the final score for the SV_tO using a decision tree. We have assumed that if the score(SV_tO) is above a certain threshold, it is a "good" SVO. Otherwise, we consider the case that this particular SV_tO might not have occurred enough times in the data set, but might still be "good". So, we take into account the scores of the corresponding pairs of SV and VO.

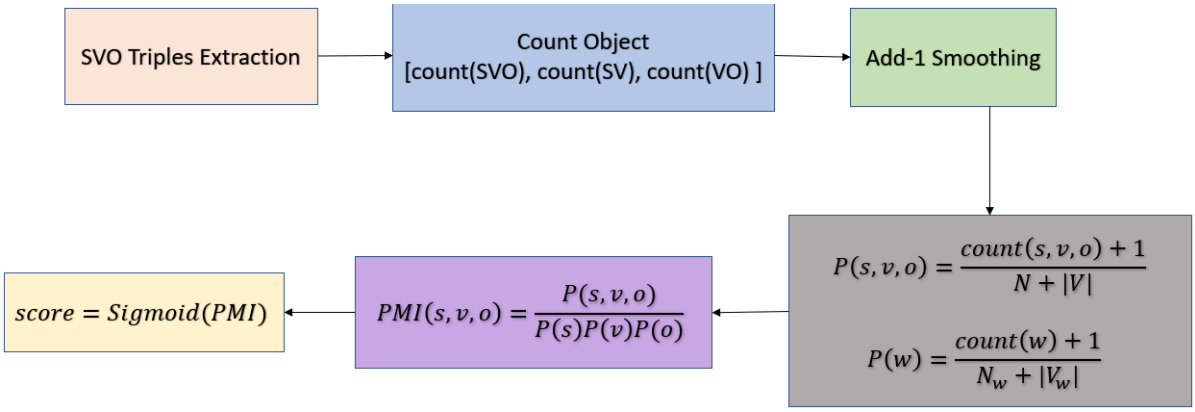

Figure 2: Good English Model

Decision Tree:

$$FinalScore(SV_tO) = \frac{score(S,V_t,O) - score(S,V,O)}{score(S,V_t,O) + score(S,V,O)}$$

If:

$$score(S,V_t,O) > score(S,V,O) + \alpha \times score(S,V,O)(1 - score(S,V,O))$$

\\where α is a model parameter

return FinalScore

else:

$$FinalScore = FinalScore + \frac{score(S,V_t) - score(S,V)}{score(S,V_t) + score(S,V)}$$

$$FinalScore = FinalScore + \frac{score(V_t,O) - score(V,O)}{score(V_t,O) + score(V,O)}$$

return $\frac{FinalScore}{3}$

Figure 3: Decision Tree

The model attempts to weight a possible sensible composition that it has not seen in the corpus before, given that it has seen its basic composing words. For example, let us say that the model has never seen the SVO: ”boy eats cake”. However, it has seen the sentences: ”woman eats cake” and ”boy eats bread”. If we consider only the contribution of SVO and SV_tO scores, this sentence will be marked as ”bad”. So, we include the score for ”boy eats” and ”eats cake”. This will increase the score of the unseen SV_tO. The decision tree for the $FinalScore$ is given in Figure 3.

The good english model only stores a count object for SVO, SV and VO. So, it is possible to update the model on the fly as it sees more ”good” SVO triples. Also, the model’s computations are completely parallelizable.

3.3 Advanced Good English

The Advanced Good English model is a modified version of the original Good English model. The Good English model looks only at the counts of the SVO, SV and VO and hence is able to capture information from the context words. But every word can have multiple senses and the sense

depends on the context. In other words, we want to give different weights to different senses.

Another aspect that needs to be captured is the inherent bias in the datasets. For example, the liking of vehicles is generally attributed to males and thus sentences like "He likes trucks" is likely to occur number of times but sentences like "She likes trucks" may never come. These sentences make sense and hence the task of giving weights differently to every sense is not an easy task of word sense disambiguation. Also, we may have never encountered the word "car" in different contexts but suppose not with liking. But we know that truck and car are related and hence need to give extra weight to sentences like "He likes cars" even though it was never seen.

For this, we use different relations that can be found between senses. We only look at synonyms, hyponyms and hypernyms. We want to give extra weight according to the overlap between of the target verb and landmark verb. We modify the formula for probability.

$$P(w) = \frac{count(w) + k + R_w}{N_w + k|V_w| + \Phi_w}$$

Here w can be SVO, SV or VO type quantity. We do Add-k smoothing and R_w is the extra weight that will be added. Φ_w is a factor that accounts the change in the denominator as the numerator will change due to R_w. The idea is to increase the count if there according to the overlap of senses and during calculation of the probabilities, we need to account for these extra counts.

3.3.1 Sense based Smoothing

We want the quantity R_w to capture a score for semantic compositionality of unseen cases. In some sense, we want to shift weights of from non-zero counts to the other possible ways of forming SVO, SV and VO, taking into consideration the semantic sense. Hence we do a smoothing that takes information of sense into consideration.

For this we first list down the nouns, N_l, and verbs, V_l, in the corpus. Any noun can appear as subject or object and thus we can form $|N_l| \times |V_l| \times |N_l|$ different SVO. We only enumerate SVO obtained by replacing S or O by all elements in N_l or V by verbs in V_l in the $SVOs$ that have non-zero count in the corpus. Let us represent the set of enumerated SVO by SVO_l. This set does not contain the SVO that have non-zero counts.

We calculate a quantity ω_N for a given SVO and NVO where N is a noun from N_l. We note down the senses of N and S and then we find the synonyms, hypernyms and hyponyms for each sense. Then we find the sense for which the Jaccard similarity of the set of synonyms is maximum. The similarity is taken to be zero if anyone of them has no synonyms (this arises because the word itself is not considered). The same thing is done for hypernyms and hyponyms (the maximum may come at for different senses). This gives us three similarity scores j_{syn}, j_{hyp} and j_{hypo}. The reason why we allow maximization on different senses is because we don't know which sense is more important i.e. we simply cannot declare some sense to be important than others and not consider others. But we know that the information given by j_{syn} is more likely to be important. Hence, we take a weighted sum of these scores to obtain ω_N.

$$\omega_N = \alpha_1 j_{syn} + \alpha_2 j_{hyp} + \alpha_3 j_{hypo}$$

We can now say that the count of NVO is given by ω_N. But the problem is that the S is not a single word but a phrase. To account for this problem, we calculate ω_N for each word in the subject phrase. Then, we take a weighted average of the obtained $\omega_N s$ where the weights are the reciprocals of the count of the word in the corpus. This is done so as to reduce the effect of words that occur large number of times and do not convey specific information about the subject. The obtained final score is taken as the count of NVO. Similarly, the count of SVN and $S\nu O$ is calculated by replacing nouns and verbs from N_l and V_l.

3.3.2 Calculation of R_w and Φ_w

We do not store any R_w. It needs to be calculated when we need to find the probability. Say that we get an SVO, in which S, V and O are phrases.

$$R_{SVO} = \sum_{N \in N_l} \omega_{NVO} + \sum_{N \in N_l} \omega_{SVN} + \sum_{\nu \in V_l} \omega_{S\nu O}$$

where:
ω_{NVO} = Count of NVO
ω_{SVN} = Count of SVN
$\omega_{S\nu O}$ = Count of $S\nu O$

The R_{SVO} increases the weights of the SVO that have more semantically sensible SVO related to it. We can similarly calculate R_{SV} and R_{VO}.

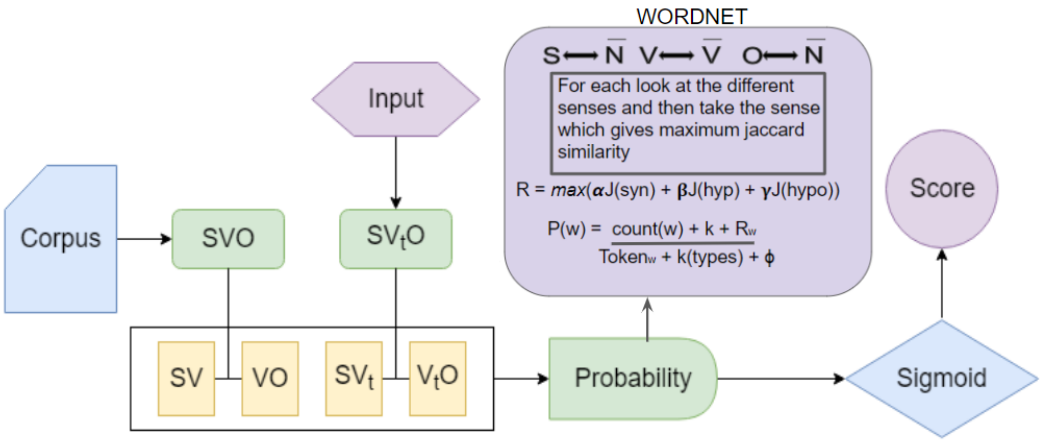

Figure 4: Advanced Good English Model

The value of Φ_w is calculated initially by calculating the R_w values on the whole corpus. The sum of these R_w values becomes Φ_w and it is stored.

3.3.3 Scoring, Model Parameters and On-the-Fly learning

For the scoring, we use the score function defined in Good English model. But here, rather than a decision tree, we have the following function for scoring:

$$FinalScore = \sum_{w \in \mathbb{P}} \beta_w \sigma(score(w)) score(w)$$

$$\mathbb{P} = \{SVO, SV, VO, SVtO, SVt, VtO\}$$

Here σ is the sigmoid function. It should be noted that while calculating the probabilities for S, V and O need to be done by:

$$P(w) = \frac{count(w) + k}{N_w + k|V_w|}$$

where N_w is $|N_l|$ for S and O and $|V_l|$ for V.

The parameters of the model include $\vec{\alpha}$, $\vec{\beta}$ and k where $\vec{\alpha}$ is for weighting the importance of synonyms, hyponyms and hypernyms. During the On-the-fly learning, we update the Φ values, updating the counts, and if necessary adding new entries. The noun list and verb list is also updated. We need not calculate the Φ values as we can directly find the R_w values for the SVO and directly add them.

3.4 BERT Model

BERT (Bidirectional Encoder Representations from Transformers) (Devlin et al., 2019) model is the state of the art in most of the Natural Language Processing applications. To perform the task of verb-replacement using BERT, we have used the pre-trained Masked LM BERT model (Alfaro et al., 2019).

The basic building blocks of BERT are transformer and attention mechanism. It is a multi-layer bidirectional transformer encoder. It has around 110 million parameters. It is trained on a large amount of BooksCorpus and Wikipedia. It can be fine-tuned for a specific task.

The verb in the input sentence is masked (verb replaced by word MASK) and both the original and masked sentences are given as an input to the model. The motive behind giving the original sentence as input is to make model understand the exact meaning of the verb. Then, the model attempts to predict a word in place of the masked word.

4 Datasets

The dataset used for the training of the models (except BERT) is **British National Corpus (BNC) Baby**. This dataset contains four million words in total, extracted from the British National Corpus. BNC Baby is in XML format and the Part of Speech tagging is already done. It contains material from the four domains-academic writing, imaginative writing, newspaper and spontaneous conversation. The dataset contains 1,68,000 total number of SVO triples.

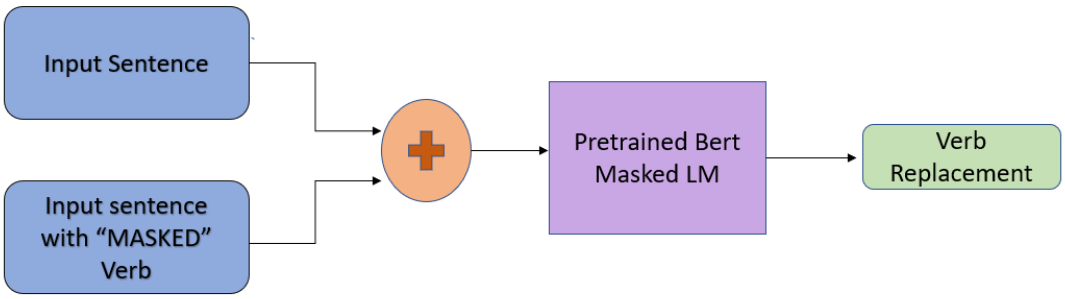

Figure 5: BERT Model

We have lemmatized the data before constructing the latent factors and core tensor. We have analyzed the data at paragraph level. The extraction of SVO triples is done using two methods:

1. Using Textacy Library
2. Using Advanced SVO extraction Method (Kashyap et al., 2019)

The dataset we used for evaluation is Grefenstette and Sadrzadeh (2011a) dataset.The data consists of 2500 simple sentences consisting of single subject-verb-object pair and a replacement verb for each sentence. The dataset also contains scores (between 0-7) given to each sentence provided by 25 participants. These scores are representation of the similarity judgement between the target verb and replacement (landmark) verb.

5 Evaluation

We are evaluating our models using the dataset made by Grefenstette and Sadrzadeh Compositional for testing compostitional meaning of sentences(Grefenstette and Sadrzadeh, 2011)[1]. Their evaluation data set has a set of SVO triples. There are multiple target verbs for each. We comp ose both the actual verb and the landmark verb with subject and object, to form two small compositional phrases. Human annotators have given a score in the scale of 1-7 for each verb replacement. For every SVO, we have calculated the average score given by the human annotators for each target replacement verb. Concurrently, we have the similarity score generated by our model for the verb replacements. We have calculated the Spearman's Rank Correlation (ρ) between the list of human annotated scores for every SVO triple's verb replacements, and the corresponding scores generated by our models. As the final metric for the model, we have taken the mean ρ for all the SVO triples in the evaluation data set.

[1]These datasets are available at http://www.cs.ox.ac.uk/activities/compdistmeaning/

6 Results

The results (Table 2) show that Good English Model and BERT give better results than Tensor Model.

7 Semantically Sensible Thesaurus

Looking for synonyms in a thesaurus is not efficient because the same word can have different implications when placed in different sentences. Semantically Sensible Thesaurus(SST) is a thesaurus that uses the mentioned models to find verb replacements. As the models take context into consideration, they are able to order the different senses of the verb and provide verbs that fits the meaning of the sentence rather than giving a complete list of synonyms.

For the case of Tensor Model and Good English Model, we extract the SVO and then by treating the synonyms of V as possible candidates for replacement, we obtain the scores of these verbs. In case of BERT, the whole sentence is given with the verb masked.

A few examples:

Input	**Suggested**
Horse eats grass	Horse consumes grass
Computer runs the program	Computer executes the program
I buy this idea	I like this idea

Annotator	Verb	Subj	Obj	Target Verb	Score
20	run	family	hotel	operate	7
20	run	family	hotel	move	2

Table 1: An example of the scores given in the evaluation data set

	Textacy	Advanced SVO Extractor
Tensor Model	0.07	0.01
Good English	0.40	0.15
Advanced Good English	0.278	0.13
BERT	0.147	

Table 2: Table showing the performance of different models

8 Conclusion

A major challenge for the tensor-based factorization model as well as both the Good English models, is to find a reliable method for the extraction of SVO triples. We observed a change in performance of all the models, when we computed the SVO triples using two different methods. We have concluded that SVO extraction is the bottleneck of these models. If we are able to find a more efficient method of SVO extraction, we expect the model performance to increase significantly.

From the results, it seems that the model based on BERT performs worse than the rest of the models even though it does not rely on this bottleneck of SVO extraction. This is because the evaluation is happening on a data set for which correct SVOs have already been provided. In real scenarios, when we input a sentence like in the case of the Semantically Sensible Thesaurus, BERT takes the sentence as the input directly. Whereas, the other models are given the SVOs extracted from the sentence. Since, these SVOs are not reliable, the models do not give as good results as expected. The results of BERT appear to be much better in their comparison, by manual observation.

Although Good English Model and Advanced Good English Model rely on the extraction of SVO triples, there is still scope for improving their performance. They have very little computation time and hence, can be updated online. New instances of "good" SVOs can be added to the model by updating just the Count Object. As the model sees new sentences, it will be able to capture more variance and multi-way interactions between words.

Advanced Good English is not able to perform better on the test data compared to the simple Good English Model because it is trying to generalize on a higher level and thus require larger corpus. Also, it heavily relies on Wordnet (Miller, 1995), which is also limited to a certain extent. Advanced Good English give better results if we manually input the sentence.

9 Future Work

In future, we want to give replacement for parts-of-speech other than verbs without disturbing the semantic sense of the sentence. We also want to use ConceptNet (Liu and Singh, 2004) in place of WordNet and extend from just English to other languages as well. Next will be to compare the results of Advanced Good English when it is given a larger corpus and possibly try to find a better method for SVO extraction.

Languages evolve over time and we are starting to see new words like "tweeting", "texting" and as Good English Model can learn online, it will be interesting to see whether it is able to capture the context and meaning of new words. At the same time, people use multiple languages while chatting. We want to see if the model is able to understand such lingua franca and give replacements accordingly and thus look at the possibility of "One Model for all Languages."

References

Felipe Alfaro, Marta R. Costa-jussà, and José A. R. Fonollosa. 2019. BERT masked language modeling for co-reference resolution. In *Proceedings of the First Workshop on Gender Bias in Natural Language Processing*, pages 76–81, Florence, Italy. Association for Computational Linguistics.

Marco Baroni and Roberto Zamparelli. 2010.

Nouns are vectors, adjectives are matrices: Representing adjective-noun constructions in semantic space. In *EMNLP*.

Bob Coecke, Mehrnoosh Sadrzadeh, and Stephen Clark. 2010. Mathematical foundations for a compositional distributional model of meaning. *ArXiv*, abs/1003.4394.

Tim Van de Cruys, Thierry Poibeau, and Anna Korhonen. 2013. A tensor-based factorization model of semantic compositionality. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1142–1151, Atlanta, Georgia. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*.

Edward Grefenstette and Mehrnoosh Sadrzadeh. 2011. Experimental support for a categorical compositional distributional model of meaning. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1394–1404. Association for Computational Linguistics.

Raiyani Kashyap, ¸calves Teresa, Gon, Quaresma Paulo, and Nogueira Vitor, Beires. 2019. Automated event extraction model for linked portuguese documents. In *Proceedings of the Text2StoryIR'19 Workshop, Cologne, Germany*.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *EMNLP*.

H. Liu and P. Singh. 2004. Conceptnet — a practical commonsense reasoning tool-kit. *BT Technology Journal*, 22(4):211–226.

George A. Miller. 1995. *WordNet: A Lexical Database for English*.

Jeff Mitchell and Mirella Lapata. 2008. Vector-based models of semantic composition. In *In Proceedings of ACL-08: HLT*, pages 236–244.

Jeff Mitchell and Mark Steedman. 2015. Orthogonality of syntax and semantics within distributional spaces. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1301–1310, Beijing, China. Association for Computational Linguistics.

Xipeng Qiu and Xuanjing Huang. 2015. Convolutional neural tensor network architecture for community-based question answering. In *IJCAI*.

Richard Socher, Brody Huval, Christopher D. Manning, and Andrew Y. Ng. 2012. Semantic compositionality through recursive matrix-vector spaces. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1201–1211, Jeju Island, Korea. Association for Computational Linguistics.