DISCOVERING SPECIFIC SEMANTIC RELATIONS AMONG WORDS USING NEURAL NETWORK METHODS

A Thesis Submitted to
the Graduate School of Engineering and Sciences of
İzmir Institute of Technology
in Partial Fulfillment of the Requirements for the Degree of

DOCTOR OF PHILOSOPHY

in Computer Engineering

by
Erhan Sezerer

October 2021
İZMİR

ACKNOWLEDGMENTS

I would like to thank my advisor Asst. Prof. Dr. Selma Tekir for her guidance throughout many years for this thesis and many other research projects and for helping to shape the early years of my career.

I would like to thank my PhD committee members Assoc. Prof. Dr. Mustafa Özuysal and Prof. Dr. Oğuz Dikenelli for their valuable feedback and guidance throughout this thesis.

I would also like to thank Assoc. Prof. Dr. Yalın Baştanlar and Assoc. Prof. Dr. Derya Birant for taking part in my PhD examination and providing valuable feedback and criticism.

The Titan V used for the experiments in this thesis is donated by the NVIDIA Corporation.

ABSTRACT

DISCOVERING SPECIFIC SEMANTIC RELATIONS AMONG WORDS USING NEURAL NETWORK METHODS

Human-level language understanding is one of the oldest challenges in computer science. Many scientific work has been dedicated to finding good representations for semantic units (words, morphemes, characters) in languages. Recently, contextual language models, such as BERT and its variants, showed great success in downstream natural language processing tasks with the use of masked language modelling and transformer structures. Although these methods solve many problems in this domain and are proved to be useful, they still lack one crucial aspect of the language acquisition in humans: Experiential (visual) information.

Over the last few years, there has been an increase in the studies that consider experiential information by building multi-modal language models and representations. It is shown by several studies that language acquisition in humans start with learning concrete concepts through images and then continue with learning abstract ideas through text. In this work, the curriculum learning method is used to teach the model concrete/abstract concepts through the use of images and corresponding captions to accomplish the task of multi-modal language modeling/representation. BERT and Resnet-152 model is used on each modality with attentive pooling mechanism on the newly constructed dataset, collected from the Wikimedia Commons. To show the performance of the proposed model, downstream tasks and ablation studies are performed.

Contribution of this work is two-fold: a new dataset is constructed from Wikimedia Commons and a new multi-modal pre-training approach that is based on curriculum learning is proposed. Results show that the proposed multi-modal pre-training approach increases the success of the model.

ÖZET

YAPAY SİNİR AĞI YÖNTEMLERİ İLE SÖZCÜKLER ARASI ÖZEL ANLAMSAL İLİŞKİLERİN KEŞFEDİLMESİ

Doğal dillerin anlaşılması, bilgisayar bilimlerinin en eski problemlerinden biridir. O günlerden bu yana, birçok çalışma dillerdeki anlamsal birimlerin (kelime, hece ve harf) temsiline adanmıştır. Yakın zamanda, BERT ve türevleri gibi bağlamsal dil modelleri, maskelenmiş dil modelleme ve transformer yapıları kullanarak büyük başarılar göstermiştir. Bu metodlar, alandaki birçok problemi çözmesine ve kullanışlılığını kanıtlamasına rağmen dil öğreniminde önemli bir rolü olan deneyimsel (görsel) bilgiyi dikkate almamaktadır.

Son birkaç yılda deneyimsel bilgiyi göz önünde bulunduran çok-kipli dil modelleri ve temsilleri üzerine olan çalışmalarda artış vardır. Birkaç çalışmanın gösterdiği üzere, dil öğrenimi insanlarda imgelerden somut kavramları öğrenerek başlar ve yazım yoluyla soyut kavramları öğrenerek devam eder. Bu çalışmada, somut kavramları imgeden öğrenen ve soyut kavramları yazımdan öğrenen, izlence öğrenimi yöntemini kullanan bir çok-kipli dil modeli/temsili önerilmiştir. Yazım ve imge kipleri için sırasıyla BERT ve Resnet-152 modelleri, dikkat havuzlaması yöntemiyle biraraya getirilerek, yeni oluşturulmuş Wikimedia Commons veri kümesi üzerinde kullanılmıştır. Önerilen metodun başarımı doğal dil işleme görevleri üzerinde ve ablasyon çalışması ile sınanmıştır.

Bu çalışmanın katkısı iki yönlüdür: Wikimedia Commons kullanılarak yeni bir veri kümesi oluşturulmuş ve izlence öğrenimine dayanan yeni bir çok-kipli ön-eğitim yaklaşımı önerilmiştir. Elde edilen sonuçlar bu çok-kipli ön-eğitim yönteminin modelin başarımını artırdığını göstermektedir.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

Understanding human languages has always been an important sub-challenge towards intelligent machines. The effort of creating a language model and grasping the meaning of words and sentences is almost as old as the computer science itself and can be traced back to finite automata and Turing Machines. Since then, decades of scientific work are made in order to find a good way of representing the meaningful units in languages such as words, lexemes, and morphemes.

Initial works towards this goal showed an interest in representing the words as discrete and independent one-hot vectors where each entry corresponds to a word in dictionary. This orthogonal representation of words leads to an implicit assumption that the words are independent from each other in meaning and distance. This leads to a system where the words "dog, swimsuit" and the words "dog, cat" have similar distance and difference in meaning. As a way of solving this problem, researchers introduced the count-based language models, which are also called n-gram models (Chen and Goodman (1996), Kneser and Ney (1995)), that rely on the distributional hypothesis (Harris (1954)). In an n-gram model, words are represented with row vectors from a word-word co-occurrence matrix. Although this method solves the independence problem and create a way for measuring and representing the similarity among words, it still had two major drawbacks. First is the inability to consider word order. Although word n-grams can compute the possibility of words and their relations, they only do so in a bag-of-words fashion where word order is ignored. Second major drawback is called the curse of dimensionality. There are billions of different n-grams that can occur in any language and it is highly unlikely to see most of the n-grams in training sets. For example, The Oxford English dictionary has $171,476$ words which leads to a bi-gram count of $29,404,018,576$. Considering that the entire Wikipedia dataset has 3.78 billion words, most of the bi-grams would remain unseen during the training phase.

Neural network-based language models (Hinton et al. (1986), Elman (1990), Bengio et al. (2003)) emerged as a solution to these problems. Early neural network models mostly trained with next word prediction and use the weights of hidden layer as representation of words. This method allows the words to be represented by dense distributed vectors which solves all of the problems of n-gram models mentioned above. They account for word order since they use next word prediction as the training objective and they solve the

curse of dimensionality because the representations of words come from the same hidden layer. This means that the similar words will trigger the same type of output from the system therefore bringing all of the words with similar meaning together even if they are not seen in the training set. For example, even if the system does not see the sentence "dog was running in the bedroom" in the training set, it can still learn the meanings if it sees the sentence "cat was running in the bedroom". This is due to the fact that dogs and cats are both animals and pets and therefore will be very close to each other in the vector space.

The breakthrough in word representations with neural networks drove the research into focusing on specific aspects of representing words and their similarities such as sense discrimination (Schütze (1998), Reisinger and Mooney (2010), Huang et al. (2012)), identifying antonyms and synonyms (Nguyen et al. (2016), Yu et al. (2015)), representing morphemes (Luong et al. (2013)) and many more. Although each such subtopic has seen a substantial improvement, there was a lack of a model that combines each such property into a single model and a single training scheme. The attempt of building a model that encompasses all of these aforementioned properties led to algorithms with contextual representations (Melamud et al. (2016), Howard and Ruder (2018), Peters et al. (2018), Devlin et al. (2019)).

Instead of considering words as entities in a dictionary of word embeddings, contextual representation models considered each word as the aggregate of the words around them in a particular sentence. Therefore, each context can provide different embedding/meaning for words, automatically capturing the sense and semantic relations such as hypernymy/hyponymy and antonymy/synonymy. Various methods have been proposed to create contextual models. Initially, research in this area (such as Melamud et al. (2016) and Howard and Ruder (2018)) focused on using bidirectional LSTMs (Hochreiter and Schmidhuber, 1997) with next word prediction. Later, this focus is shifted towards transformer models (Vaswani et al., 2017) starting with BERT (Devlin et al., 2019) and its variants (RoBERTa (Liu et al., 2019), AlBERT (Lan et al., 2020)). Contextual models showed great success in many downstream tasks by exceeding human baselines in some of the tasks while performing very close to them in others.

Although text-based language models show human-level performance in many tasks, there have been studies that show language acquisition in children can mostly be attributed to experiential information in early ages (Griffiths et al. (2007), Vigliocco et al. (2009), Andrews et al. (2009)). It is mentioned in those work that the language acquisition in children start with experiential information where we mostly learn about concrete concepts in languages, and continue with textual information in later ages where we mostly learn about abstract concepts. Thus, many researchers tried to build language models with multi-modal information (Lu et al. (2019), Agrawal et al. (2018), Anderson et al. (2018), and many more), leveraging both textual and visual inputs.

In this work, the aim is to create a multi-modal language model that uses both

textual and visual features, similar to what humans do. A neural network model is fine-tuned using abstract/concrete concepts on each part of the model, textual and visual, to reach the goal. Driven by the motivation of building a language model that mimics the human language acquisition process, curriculum learning (Elman, 1993; Bengio et al., 2009) is used to meaningfully order the training samples from easy to hard throughout the training using concrete and abstract samples as difficulty measure.

Contribution of this work is two-fold: First, a new multi-modal dataset is constructed from Wikimedia Commons. The proposed dataset is comparable in size to the largest multi-modal dataset available in literature and greater in size than most of the other ones. In addition, unlike all the other available datasets, the proposed dataset includes concreteness ratings associated with each sample as an additional feature. Second, a new multi-modal pre-training approach that is based on curriculum learning (Bengio et al., 2009) is proposed. Proposed multi-modal language model is trained with a meaningfully ordered training set, starting with concrete (easy) samples and then switching to abstract (hard) samples. This serves as a way of mimicking the learning behaviour of human language acquisition process. To our knowledge, this is the first work that uses a curriculum learning approach on multi-modal language models.

Results show that the proposed multi-modal pre-training method increases the success of the model in downstream tasks. Also, it can be seen from the ablation study that this increase in performance is consistent among all fusion techniques used in this work. Best results are obtained when the multi-modal pre-training scheme is used with attentive pooling as fusion mechanism. In addition to the tests mentioned above, several tests are performed for measuring the informativeness of the newly constructed dataset.

Rest of the thesis is structured as follows: In Chapter 2, background information is given on the task of language modeling. Model details and the new dataset are explained in Chapter 3. Experimental results are shared in Chapter 4 along with the descriptions of the datasets used. And finally in Chapter 5, final remarks are made with possible future directions.

CHAPTER 2

BACKGROUND

Finding good representations for words has always been one of the major goals in natural language processing (NLP) since almost all of the downstream tasks in NLP such as machine translation, text classification, question answering, etc. requires well-learned representations of words as an input to work successfully.

The oldest NLP systems feed one-hot vectors as word representation to the models where each word is represented with a unique vector with one of its index 1 (corresponding to the specific word) and the rest of the indices are set to zero. i.e:

$$dog = \{0, 0, ..., 0, 0, 1, 0, ..., 0\}$$
$$cat = \{0, 0, ..., 0, 1, 0, 0, ..., 0\}$$
$$bank = \{0, 0, ..., 1, 0, 0, 0, ..., 0\}$$

Orthogonality of the word vectors provides independence assumption between words which in turn helps machine learning algorithms to solve the problems without any inherent assumption on the relation of words. On the other hand, this independence assumption leads to an important disadvantage: all the words have equal distance from each other which means that the semantic relation of any two particular words is the same as the semantic relation of them with a third one. This lead to an incorrect semantic representation that makes the relation between dog and cat the same as the words dog and bank. Any NLP model that is expected to work successfully should be able to differentiate the semantic/syntactic relation of word pairs instead of assuming they are all the same.

The first solution to this problem was the count-based methods, also called the n-gram models (Chen and Goodman (1996), Kneser and Ney (1995)). N-gram models build a matrix of frequencies of each word in the dictionary with other words such as the one shown in Table 2.1 (example of a 2-gram);

Each row in this matrix is taken as the representation of corresponding words. The similarity between the words is then calculated from the similarity between their corresponding rows (vectors). N sized windows are used on the words as the context in the n-gram models (hence the name n-gram), but two specialized methods , called LDA (Blei et al. (2003)) and LSA (Deerwester et al. (1990)), took this idea further by considering topics as the context of words and by creating denser representations.

There are also generative probabilistic topic models that aim at creating word rep-

Table 2.1: bi-gram representation of "The brown fox jump over the lazy dog"

	the	brown	fox	jump	over	lazy	dog
the	0	1	0	0	0	1	0
brown	0	0	1	0	0	0	0
fox	0	0	0	1	0	0	0
jump	0	0	0	0	1	0	0
over	0	0	0	0	0	1	0
lazy	0	0	0	0	0	0	1
dog	0	0	0	0	0	0	0

resentations (Vigliocco et al. (2009), Griffiths et al. (2007), Andrews et al. (2009)). They also propose that image plays an important role in language acquisition along with the text. Therefore, they include experiential (visual) information to create word representations.

Before neural representation learning, representations of words or documents are computed using such Vector Space Models (VSM) of semantics. Turney and Pantel (2010) provide a comprehensive survey on the use of VSM for semantics. Although these count-based representations (VSMs) are proved useful in addressing semantics, they are bag-of-words approaches and are not able to capture both syntactical and semantic features at the same time, which is required for performing well in NLP tasks.

2.1 Distributional Hypothesis

The idea of building word representation from such frequency statistics comes from the "Distributional Hypothesis" (Wittgenstein (1953), Harris (1954)). Distributional hypothesis states that the meaning of a word can be determined through the words that co-occur with it in the same context. Famously, Harris (1954) states that the "words that occur in the same context tend to have similar meanings". He states that at least certain aspects of meaning are due to distributional relations. For instance, synonymy between two words can be defined as having almost identical environments except chiefly for glosses where they co-occur, e.g. oculist and eye-doctor. The author also suggests that sentences starting with a pronoun should be considered the same context as the previous sentence where the subject of the pronoun is given since their occurrence is not arbitrary and the fullest environmental unit for the distributional investigation is the connected discourse structures of such sentences.

2.2 Distributional Representations

Although the count-based methods can leverage the distributional model to learn the representation of words, they suffer from several drawbacks:

- lack of word order: n-gram models only look for the frequency of the words while disregarding the order in which they appear, which can affect their meaning (i.e. fish stick refers to the food of British cuisine, while stick fish refers to a type of tropical fish species).
- unable to retrieve representations from partial information (generalization power): Humans are able to retrieve memory from a given partial information, but n-gram solutions lack this property by making the information available only if the vector and n-grams are perfectly given.
- curse of dimensionality: they create millions, if not trillions, of different possible n-grams which are very unlikely to be observed in the training data. This will lead to a very sparse matrix with a lot of uninformative zero entries.

Neural network solutions emerged to solve these issues. In first such attempt, Hinton et al. (1986) utilize the idea of distributed representations for concepts. They propose to use patterns of hidden layer activations (which are only allowed to be 0 or 1) as the representation of meanings instead of representing words with discrete entities such as the number of occurrences together. They argue that the most important evidence of distributed representations is their degree of similarity to the weaknesses and strengths of the human mind. Unlike computer memory, the human brain is able to retrieve memory from partial information. Distributed representations conform to this notion better than local distributions (i.e. bag of words model, where each meaning is associated with a single computational unit) since the meaning of a word is distributed across several units and a loss of an activation will only slightly affect the memory retrieval process. The rest of the activations that are still there will be able to retrieve the memory. Even if the occlusion of activations is strong enough to lead the system to an incorrect meaning, it will still result in a meaning close to that of the target word, such as instead of apricot the word peach is recalled. The authors state that this phenomenon further reinforces the idea of being similar to the human mind by showing the similarities with deep dyslexia that occurs in adults with certain brain damage.

2.3 Language Modeling

Language modelling is the task of predicting the next word given a sequence of words. Formally, it is the prediction of the next word's probability distribution given a sequence of words (Equation 2.1).

$$P(x_{t+1}|x_t, ..., x_1) \tag{2.1}$$

In an alternative interpretation, a language model assigns a probability to a sequence of words. The probability calculation can be formulated as the product of conditional probabilities in each subsequent step having the assumption that they are independent (Equation 2.2).

$$\begin{aligned} P(x_1, ..., x_t) &= P(x_1)P(x_2|x_1)P(x_3|x_2, x_1)...P(x_t|x_{t-1}, ..., x_1) \\ &= \prod_{i=1}^{t} P(x_i|x_{i-1}, ..., x_1) \end{aligned} \tag{2.2}$$

In traditional language modelling, the next word's probability is calculated based on the statistics of n-gram occurrences. n-grams are n consecutive words. In n-gram language models (Chen and Goodman (1996), Kneser and Ney (1995)), an n-gram's probability is computed depending on the preceding $n - 1$ words instead of using the product of conditional probabilities of bi-grams, tri-grams, etc. to simplify the computation.

N-gram language models have some issues. When the length of n-grams increases, their occurrence becomes sparse. This sparsity causes zero or division by zero probability values. The former one is resolved by smoothing and back-off is used to deal with the latter. Sparsity provides coarse-grained values in the resultant probability distribution. Moreover, storing all n-gram statistics becomes a major problem when the size of n increases. This curse of dimensionality is a bottleneck for n-gram language models.

2.4 Distributional Representations Through Language Modeling

Elman (1990) was the first to implement the distributional model proposed by Hinton et al. (1986), in a language model. He proposes a specific recurrent neural network structure with memory, called the Elman network, to predict bits in temporal sequences. Memory is provided to the network through the use of context units that are fully connected with hidden units (Figure 2.1).

To show that distributional representations can be learned through language modelling, he makes two different experiments, the first of which involves a simulation to

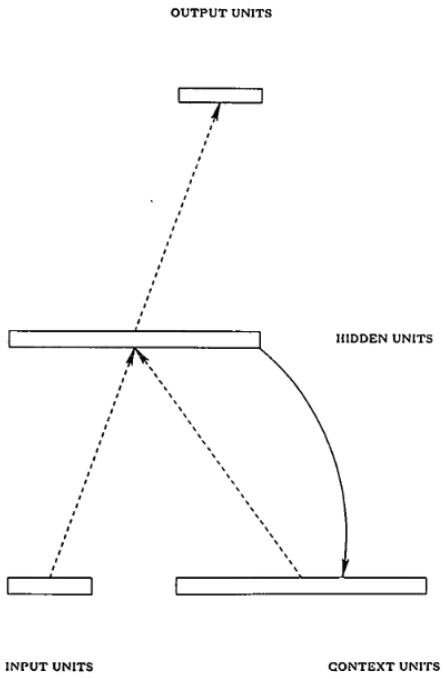


Figure 2.1: Elman Network (Elman (1990)).

predict bits in the XOR problem. The input sequence is in the form of an input pair followed by an output bit. In the solution scheme, two hidden units are expected to represent two main patterns in the XOR truth table. That is one hidden unit should have high activation for 01 or 10 patterns and the other should recognize 11 or 00 patterns. In his second experiment, as an alternative problem, letter sequences that are generated partially random and partially by a simple rule are tried to be learnt by a recurrent neural network where hidden unit activations are used to represent word meanings. The idea is that by using such network structures, time can be modelled implicitly. In other words, the use of a recurrent neural network helps in learning temporal structure in language.

Xu and Rudnicky (2000) create the first language model based on neural networks. Their proposed model is based on a single fully connected layer and uses one-hot vectors of words as inputs and outputs. They highlight computational cost as the major problem and in tackling the issue they mention the necessity of update mechanisms that only update those weights with non-zero input value due to one-hot encoding.

Although these models build the theoretical and practical basis of neural word representations, Bengio et al. (2003) popularize the distributional representation idea by realizing it through a language model and lead to numerous other studies that are built on it. In their model architecture, they use a feed-forward network with a single hidden layer and optional direct connections from the input layer to the softmax layer (Figure 2.2).

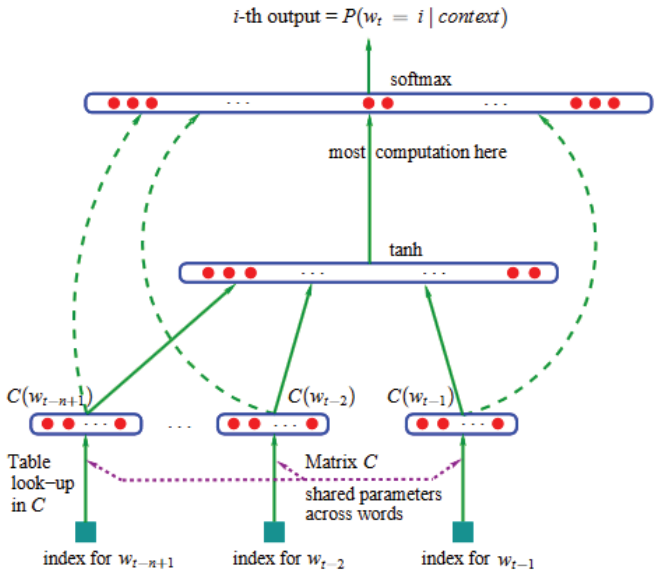


Figure 2.2: Neural Network Architecture in Bengio et al. (2003).

Weights of the hidden layer are then taken as a representation of words such as:

$$dog = \{0, 345, 0, 751, ..., -0.621\}$$
$$cat = \{0.931, 0.003, ..., 0.169\}$$
$$bank = \{-0.621, 0.413, ..., -0.884\}$$

Cosine similarity of the above word vectors can then give us a measure of how similar the given words are which can be quite useful in downstream NLP tasks compared to one-hot vectors.

In addition to the advantages discussed by the aforementioned earlier works, they argue that distributional representations also break the curse of dimensionality in traditional n-gram models (Chen and Goodman (1996), Kneser and Ney (1995)) where the probability of each word depends on the discrete n-grams whose numbers can exceed millions. A considerably high number of such n-grams will highly unlikely to be observed in the training set which results in sparsity problems in conditional probability calculations. A real-valued feature vector representation of words will overcome this problem by working with a smooth probability function. The conditional probability of seeing a word given a context is calculated by updating the index of that word on the shared representation matrix of all the vocabulary. The probability function is smooth in that the updates that are caused by similar contexts are alike.

A second advantage of the model is the ability to capture context-based similarities. In n-gram models, the sentences "the cat is walking in the bedroom" and "a dog was running in a room" will be considered as dissimilar since they are unable to consider contexts further than $1-2$ words and have no notion of similarity among word meanings. On the other hand, in the proposed model, increasing the probability of the sentence "the

cat is walking in the bedroom" will increase the probability of all the sentences below and help us generalize better:

"a dog was running in a room"
"the cat is running in a room"
"a dog is walking in a bedroom"

2.5 Word Embeddings

Once it is shown that neural language models are efficiently computable by Bengio et al. (2003), newer language models along with better word embeddings are developed successively. Table 2.2 shows the properties of word embeddings mentioned in this section.

Alexandrescu and Kirchhoff (2006) (***FNLM***) improve the model proposed by Bengio et al. (2003) by including word-shape features such as stems, affixes, capitalization, POS class, etc. at the input.

Morin and Bengio (2005) focus on improving the performance of the earlier neural language models. Instead of using softmax and predicting the output word over the entire dictionary, they propose a hierarchical organization for vocabulary terms. A binary tree of words is created based on the IS-A relation of the Wordnet hierarchy. Instead of directly predicting each word's probability, prediction is performed as a binary decision over the constructed tree's branches and leaves. This technique is an alternative to importance sampling to increase efficiency. Although the authors report exponential speed-up, the accuracy of the resultant word embeddings is a bit worse than the original method and importance sampling.

Mnih and Hinton (2008) improve the hierarchical language model proposed by Morin and Bengio (2005) by constructing and using a word hierarchy from distributional representations of words rather than a hierarchy built out of Wordnet. Thus, their approach is entirely unsupervised. They calculate feature vectors for words by training a hierarchical log-bilinear model (***HLBL***) and apply EM algorithm on the mixture of two Gaussians to construct a data-driven binary tree for words in the vocabulary. Authors also represent different senses of words as different leaves in the tree which is proposed in Morin and Bengio (2005) but not implemented. Their model outperforms non-hierarchical neural models, the hierarchical neural language model that is based on Wordnet hierarchy, and the best n-gram models (Chen and Goodman (1996), Kneser and Ney (1995)).

Mnih and Hinton (2007) propose three different language models that use distributed representation of words. In Factored Restricted Boltzmann Machine (***RBM***), they put an additional hidden layer over the distributed representation of preceding words and

Table 2.2: Properties of word embedding models.

Model	Dimension	NN Model	Aim	Knowledge-Base(s)
Bengio et al. (2003)	100	FFNN	Training	-
Morin and Bengio (2005)	100	FFNN	Performance	Wordnet
FNLM (Alexandrescu and Kirchhoff, 2006)	45-64	FFNN	Training	LDC ECA, Turkish News
LBL (Mnih and Hinton, 2007)	100	RBM, FFNN	Training	-
HLBL (Mnih and Hinton, 2008)	100	LBL	Performance	-
C&W (Collobert and Weston, 2008)	15-100	FFNN, CNN	Training	-
RNNLM (Mikolov et al., 2010)	60-400	RNN	Training	-
CBOW (Mikolov et al., 2013)	300-1000	FFNN	Training	-
Skip-Gram (Mikolov et al., 2013)	300-1000	FFNN	Training	-
SGNS (Mikolov et al., 2013)	300	FFNN	Performance	-
ivLBL/vLBL (Mnih and Kavukcuoglu, 2013)	100-600	LBL	Performance	-
GloVe (Pennington et al., 2014)	300	LBL+coocurence Matrix	Training	-
DEPS (Levy and Goldberg, 2014)	300	CBOW	Training	Stanford tagger, Dependency parser
Ling et al. (2015)	50	CBOW+Attn.	Training	-
SWE (Liu et al., 2015)	300	Skip-Gram	Training	Wordnet
Faruqui et al. (2015)	-	-	fine-tuning	PPDB, FrameNet, WordNet
Yin and Schütze (2016)	200	-	Ensemble	-
Ngram2vec (Zhao et al., 2017)	300	SGNS+n-gram	Training	-
Dict2vec (Tissier et al., 2017)	300	Skip-Gram	Training	Oxford, Cambridge, and Collins dict.

exploit interactions between this hidden layer and the next word's distributed representation. In temporal RBM, they further put temporal connections among hidden layer units to capture longer dependencies in the previous set of words, and finally in the log-bilinear model, called ***LBL***, they use linear dependencies between the next word and the preceding set of words. They report that the log-bilinear model outscores RBM models and also n-gram models (Chen and Goodman (1996), Kneser and Ney (1995)).

Collobert and Weston (2008) and Collobert et al. (2011) (***C&W***) are among the precursors in using distributed representations in various NLP problems such as part-of-speech tagging, named entity recognition, chunking, and semantic role labelling. They propose a unified architecture for all of the problems where the words in the sentences are represented by word vectors trained from the Wikipedia Corpus in an unsupervised fashion. Although they use feed-forward architecture with a sliding window approach in word-level tasks, they utilize a convolutional neural network (CNN) architecture in semantic role labelling. This is done to incorporate the varying lengths of sentences, since, in semantic role labelling, sliding window-based approaches don't work because target words may depend on some other faraway words in a sentence. By making use of trained word vectors and neural network architecture, their proposed method can capture the meaning of words and succeed in various NLP tasks (almost) without making use of hand-crafted features. Their overall scheme is described as semi-supervised being composed of unsupervised language modelling and other supervised tasks.

Mikolov et al. (2010) propose a recurrent neural network based-language model (***RNNLM***), from where word representations can be taken. The model can work on

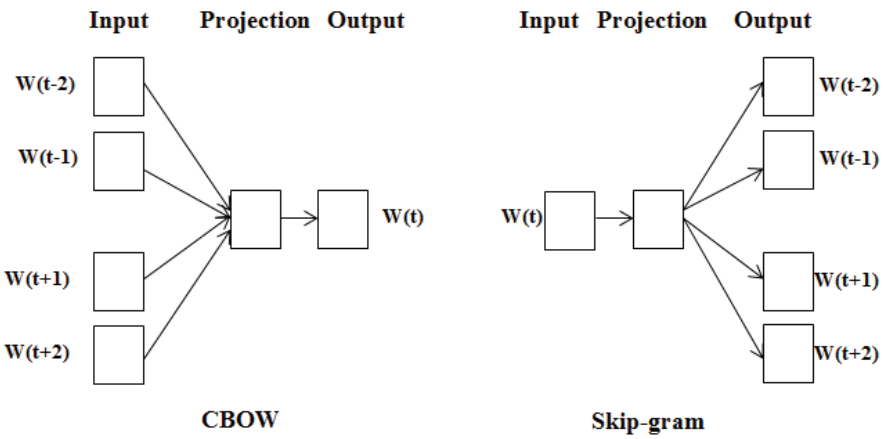


Figure 2.3: CBOW and Skip-Gram Model Architectures

contexts of arbitrary length, unlike the previous feed-forward methods where a context size should be defined beforehand. The network can learn longer dependencies. It is proved useful in tasks involving inflectional languages or languages with large vocabulary when compared to n-gram language models (Chen and Goodman (1996), Kneser and Ney (1995)).

Word2vec (Mikolov et al., 2013) is the first neural word embedding model that efficiently computes representations to leverage the context of target words. Thus, it can be considered as the initiator of early word embeddings (Tekir and Bastanlar, 2020).

Mikolov et al. (2013) propose word2vec to learn high-quality word vectors. The authors removed the non-linearity in the hidden layer in the proposed model architecture of Bengio et al. (2003) to gain an advantage in computational complexity. Due to this basic change, the system can be trained using billions of words efficiently. word2vec has two variants: Continuous Bag of Words model (CBOW) and Skip-gram model (Figure 2.3).

In ***CBOW***, the middle word is predicted given its context, the set of neighbouring left and right words. When the input sentence "nature is pleased with simplicity" is processed, the system predicts the middle word "pleased" given the left and right contexts.

Every input word is in one-hot encoding where there is a vocabulary size (V) vector of all zeros except a one in that word's index. In the single hidden layer, instead of applying a non-linear transformation, the average of the neighbouring left and right vectors (w_c) is computed to represent the context. As the order of words is not taken into consideration by averaging, it is named as a bag-of-words model. Then the middle word's (w_t) probability given the context ($p(w_t|w_c)$) is calculated through softmax on context-middle word dot product vector (Equation 2.3). Finally, the output loss is calculated based on the cross-entropy loss between the system predicted output and the ground-truth middle word.

$$p(w_t|w_c) = \frac{exp(w_c \cdot w_t)}{\sum_{j \in V} exp(w_j \cdot w_t)} \tag{2.3}$$

In ***Skip-gram***, the system predicts the most probable context words for a given input word. In terms of a language model, while CBOW predicts an individual word's probability, Skip-gram outputs the probabilities of a set of words, defined by a given context size. Due to high dimensionality in the output layer (all vocabulary words have to be considered), Skip-gram has higher computational complexi ty compared to CBOW.

To deal with this issue, rather than traversing all vocabulary in the output layer, Skip-gram with Negative Sampling (***SGNS***) (Mikolov et al., 2013) formulates the problem as a binary classification where one class represents the current context's occurrence probability whereas the other class is all other vocabulary terms' occurrence probability in the present context. In the latter probability calculation, a negative sampling method is incorporated (Mnih and Teh, 2012), which is influenced by Noise Contrastive Estimation (NCE) (Gutmann and Hyvärinen, 2012), to speed up the training process. As vocabulary terms are not distributed uniformly in contexts, sampling is performed from a distribution where the order of frequency of vocabulary words in corpora are taken into consideration. SGNS incorporates this sampling idea by replacing the Skip-gram's objective function. The new objective function (Equation 2.4) depends on maximizing $P(D = 1|w, c)$ where w, c is the word-context pair. This probability denotes the probability of (w, c) coming from the corpus data. Additionally, $P(D = 0|u_i, c)$ should be maximized if (u_i, c) pair is not included in the corpus data. In this condition, (u_i, c) pair is sampled, as the name suggests negative sampled k times.

$$\sum_{w,c} \left(\log \sigma \left(\overrightarrow{w} \cdot \overrightarrow{c} \right) \right) + \sum_{i=1}^{k} \left(\log \sigma \left(\overrightarrow{-u_i} \cdot \overrightarrow{c} \right) \right) \tag{2.4}$$

Both word2vec variants produced word embeddings that can capture multiple degrees of similarity including both syntactic and semantic regularities. The authors also made a contribution by realizing that simple algebraic operations work on the word representations. i.e. if we subtract the vector "man" from the vector "king" and add the vector "woman", the closest word in the dictionary to the resulting vector is the vector "queen".

Mnih and Kavukcuoglu (2013) introduce speedups to CBOW and Skip-gram models (Mikolov et al., 2013), called ***vLBL*** and ***ivLBL***, by using noise-contrastive estimation (NCE) (Gutmann and Hyvärinen (2012)) for the training of the unnormalized counterparts

of these models. Training of the normalized model has a high cost due to normalization over the whole vocabulary (denominator term in Equation 2.3). NCE trains the unnormalized model by adapting a logistic regression classifier to discriminate between samples under the model and samples from noise distribution. Thus, the computational cost and accuracy become dependent on the number of noise samples. With the relatively small number of noise samples, the same accuracy level with the normalized models is achieved in considerably shorter training times.

Pennington et al. (2014) combine global matrix factorization and local context window-based prediction to form a global log bilinear model called ***GloVe***. GloVe uses ratios of co-occurrence probabilities of words as weights in its objective function to cancel out the noise from non-discriminative words. As distinct from CBOW and Skip-gram (Mikolov et al., 2013), instead of cross-entropy, GloVe uses weighted least squares regression in its objective function. For the same corpus, vocabulary, window size, and training time, GloVe consistently outperforms word2vec.

Zhao et al. (2017) (***ngram2vec***) improve word representations by adding n-gram co-occurrence statistics to the SGNS (Mikolov et al., 2013), GloVe (Pennington et al., 2014), and PPMI models (Levy et al., 2015). To incorporate these statistics into the SGNS model, instead of just predicting the context words, they also predict the context n-gram of words. In order to add it to other systems, they just add n-gram statistics to co-occurrence matrix of words. They show improved scores over the models that they are built upon.

Levy and Goldberg (2014) argue that although the word embeddings with Skip-gram can capture very useful representations, they also learn from unwanted co-occurrences in the context, e.g. *Australian* and *discovers* in the sentence *"Australian scientist discovers stars with a telescope"*. In order to create a different context, they use dependency trees to link each word in the sentence to the other according to the relations they have. Their experimental results show that while their model (***DEPS***) is significantly better at representing syntactic relationships, it is worse at finding semantic relationships. In this work, they also share a non-trivial interpretation of how word embeddings learn representations, which is very rare in neural network solutions, by examining the activations of context for specific words.

Ling et al. (2015) augment CBOW (Mikolov et al., 2013) with an attention model in order to solve the shortcomings of it: inability to account for word order and lack of treating the importance of context words differently. They show that their method can obtain better word representations than CBOW while still being faster than its complementary model Skip-gram (Mikolov et al., 2013).

Yin and Schütze (2016) put forward the idea of ensembling the existing embeddings in order to achieve performance enhancement and improved coverage of the vocabulary. They propose four different ensemble approaches on five different word embeddings: Skip-Gram (Mikolov et al., 2013), Glove (Pennington et al., 2014), C&W (Collobert and

Weston, 2008), Huang (Huang et al., 2012), and Turian (Turian et al., 2010). The first method CONC simply concatenates the word embeddings from five different models. SVD reduces the dimensionality of CONC. 1toN creates meta-embeddings and 1toN^+ creates OOV words for individual sets by randomly initializing the embeddings for OOVs and the meta-embeddings, then uses a setup similar to 1toN to update meta-embeddings as well as OOV embeddings. They also propose a MUTUALLEARNING method to solve the OOV problem in CONC, SVD, and 1toN. They show that the ensemble approach outperforms individual embeddings on similarity, analogy, and POS tagging tasks.

There has also been some work to improve early word embeddings through knowledge bases.

Liu et al. (2015) (***SWE***) try to improve word embeddings by subjecting them to ordinal knowledge inequality constraints. They form three different types of constraints:

1. Synonym-antonym rule: A synonym of a word should be more similar than an antonym. They find these pair of words from the WordNet (Miller, 1995) synsets.

2. Semantic category rule: Similarity of words that belong to the same category should be larger than the similarity of words that are in different categories. i.e. (hacksaw, jigsaw) similarity should be greater than (hacksaw, mallet).

3. Semantic hierarchy rule: Shorter distances in hierarchy should infer larger similarities between words compared to long-distance cases. i.e (mallet, hammer) similarity should be larger than (mallet, tool).

The last two rules are constructed from the hypernymy-hyponymy information from Wordnet. They combine these constraints with the Skip-gram algorithm (Mikolov et al., 2013) to train word embeddings and show that they can improve upon the baseline algorithm.

Faruqui et al. (2015) aim to improve word embeddings with information from lexicons with a method called retrofitting. They use a word graph where each word is a vertex and each relation in the knowledge-base is an edge between words. In their algorithm, they bring closer the words that are shown to be connected in the word graph and words that are found to be similar from the text. In other words, while they bring closer the words related in synsets, they also preserve the similarity in the underlying pre-trained word embeddings (Skip-gram (Mikolov et al., 2013), GloVe (Pennington et al., 2014), etc.). They use various knowledge-bases such as PPDB (Pavlick et al., 2015), WordNet (Miller, 1995), and FrameNet (Baker et al., 1998).

Tissier et al. (2017) (***dict2vec***) improve word2vec (Mikolov et al., 2013) by incorporating dictionary information in the form of strong and weak pair of words into the training process. If a word a is in the definition of the word b in dictionary and b is in the definition of a too, then it is a strong pair. On the other hand, if a is in the definition

of b but b is not in the definition of a, then they form a weak pair. The authors add this positive sampling information into the training process proportional to hyperparameters.

Despite the success of these earlier word embeddings, there were still many limitations in terms of the accuracy of representations each of which is targeted by many research:

- Polysemy: All of the aforementioned models can learn only one vector for each unique word in the dictionary, however, a polysemous word can have very different meanings depending on the context they use. For example, the word "bank" can refer to a completely different entity when it is used in the context of "finance" or the context of "river". Since these models collapse all of the senses of a word in one representation, they lose the accuracy of their representation for all senses.

- Morphology: Words are not the smallest unit of language that can carry a meaning, it is the morphemes. In languages that are not morphologically rich, such as English, this is not a big issue, but in languages such as Finnish and Turkish, it becomes problematic. Some of the word forms that are constructed with suffixes, cannot be seen much during the training phase. Therefore they cannot be learned with the neural methods mentioned so far, although their meanings are indeed related to the words they are built from.

- Antonymy/Synonymy: Since both antonyms and synonyms can have similar context words, neural methods described above cannot differentiate between them successfully.

- Hypernymy/Hyponymy: Similar to the problem above hypernyms/hyponym words also share a lot of their context words, therefore making them hard to differentiate between.

In the next subsections, these limitations (such as morphology, senses, antonymy/synonymy and so on) and the proposals to their solutions are discussed.

2.6 Embeddings Targeting Specific Semantic Relations

Although initial word embedding models were successful at identifying semantic and syntactic similarities of words, they still need to be improved to address specific semantic relations among words such as synonymy-antonymy and hyponymy-hypernymy. To illustrate, consider the sentences "She took a sip of hot coffee" and "He is taking a sip of cold water". The antonyms "cold" and "hot" are deemed to be similar since their

Table 2.3: Embeddings Targeting Specific Semantic Relations.

Work	Base Model	Knowledge-Base	Target Relations
Nguyen et al. (2016)	SGNS	WordNet, Wordnik	Synonym-Antonym
Mrkšić et al. (2016)	GloVe, paragram-SL999	WordNet, PPDB 2.0	Synonym-Antonym
Vulić et al. (2017)	SGNS	✗	Synonym-Antonym
Yu et al. (2015)	✗	Probase	Hyponym-Hypernym
Luu et al. (2016)	✗	WordNet	Hyponym-Hypernym
Nguyen et al. (2017)	SGNS	WordNet	Hyponym-Hypernym
Wang et al. (2019)	Skip-gram	✗	Synonym-Antonym, Hyponym-Hypernym, Meronym

context is similar. Therefore, it becomes an issue to differentiate the synonyms "warm" and "hot" from the antonyms "cold" and "hot" considering they have similar contexts in most occurrences.

Table 2.3 presents the main approaches addressing synonym-antonym relations, hyponym-hypernym relations, and a study covering all types of relations.

Nguyen et al. (2016) propose a weight update for SGNS (Mikolov et al., 2013) to identify synonyms and antonyms from word embeddings. Their system (***dLCE***) increases weights if there is a synonym in the context and makes a reduction in the case of an antonym. To come up with a list of antonyms and synonyms, they use WordNet (Miller, 1995) and Wordnik. They report state-of-the-art results in similarity tasks and synonym-antonym distinguishing datasets.

Mrkšić et al. (2016) propose the counter-fitting method to inject antonymy (REPEL) and synonymy (ATTRACT) constraints into vector space representations to improve word vectors. The idea behind ATTRACT rule is that synonymous words should be closer to each other than any other word in the dictionary and in a similar way, REPEL constraint assumes that an antonym of a word should be farther away from the word than any other word in the dictionary. As knowledge-bases, they use WordNet (Miller, 1995) and PPDB 2.0 (Pavlick et al., 2015), and as pre-trained word vectors, they use GloVe (Pennington et al., 2014) and paragram-SL999 (Wieting et al., 2015). They report state-of-the-art results on various datasets.

Vulić et al. (2017) use ATTRACT and REPEL constraints on pre-trained word embeddings. The aim of their algorithm is to pull together ATTRACT pairs while pushing REPEL pairs apart. For forming the ATTRACT and REPEL constraints, inflectional and derivational morphological rules of four languages are used; English, Italian, Russian, and German. ATTRACT constraints consist of suffixes such as (-s, -ed, -ing) to create ATTRACT word pairs such as (look, looking), (create, created). On the other hand, REPEL constraints consist of prefixes like (il-, dis-, anti-, mis-, ir-, ..) to create REPEL word pairs such as (literate, illiterate), (regular, irregular). In order to balance the changes they make to the original embeddings (they use SGNS (Mikolov et al., 2013)), there is a third constraint that tries to pull word embeddings to their original position.

In their work, Yu et al. (2015) train term embeddings for hypernymy identification. They use Probase (Wu et al., 2012) as their training data for hypernym/hyponym pairs and impose three constraints on the training process: 1) hypernyms and hyponyms should be similar to each other (dog and animal), 2) co-hyponyms should be similar (dog and cat), 3) co-hypernyms should be similar (car and auto). They create a neural network architecture to update word embeddings without optimizing parameters. They use 1-norm distance as a similarity measure. They use an SVM on the output term embeddings to decide whether a word is a hypernym/hyponym to another word.

Luu et al. (2016) aim to identify is-a relationship through a neural network architecture. First, they extract hypernyms and hyponyms using the relations in WordNet (Miller, 1995) to form a training set. Second, they create (hypernym, hyponym, context word) triples by finding all sentences in the dataset that contain two of the hypernym/hyponyms found in the first step and using the words between the hypernym and hyponym as context words. Then, they give hyponym and context words as input to the neural network and try to predict the hypernym by aggregating them with a feed-forward neural network. The resultant hypernym, hyponym pairs along with an offset vector are given to SVM to predict whether there is an is-a relationship or not. Authors state that since their method takes context words into account, their embeddings have good generalization capability and able to identify unseen words.

Nguyen et al. (2017) aim to learn hierarchical embeddings for hypernymy. They leverage hypernymy-hyponymy information from WordNet (Miller, 1995) and propose objective functions over/above SGNS embeddings (Mikolov et al., 2013) to move hypernymy-hyponymy pairs closer. The first objective function is based on the distributional inclusion hypothesis while the second one adopts the distributional informativeness. They also propose an unsupervised hypernymy measure to be used by their hierarchical embeddings. In the proposed hypernymy measure, the cosine similarity between the hypernym and hyponym vectors (to detect the hypernymy) is multiplied by the hypernym to hyponym magnitude ratio (to account for the directionality of the relation by the assumption that hypernyms are more general terms, being more frequent and thus having a large magnitude compared to hyponyms). Their evaluation tests the generalization capability of their hypernymy solution as well, which proves that the model learns rather than memorizing prototypical hypernyms.

Wang et al. (2019) propose a neural representation learning model for predicting different types of lexical relations e.g. hypernymy, synonymy, meronymy, etc. Their solution avoids the "lexical memorization problem" because relation triples' embeddings are learned rather than computing those relations through individual word embeddings. In order to learn a relation embedding for a pair of words, they use the Skip-gram model (Mikolov et al., 2013) over the neighbourhood pairs where the similarity between pairs is defined on hyperspheres. Their lexical relation classification results verify the effectiveness

Table 2.4: Sense Embeddings.

Unsupervised			
R&M (Reisinger and Mooney, 2010)			
Supervised			
Work	Knowledge Base	Probabilistic	NN Model
Huang et al. (2012)	✗	Spherical k-means	Custom Language Model using both local and global context
Pelevina et al. (2016)	✗	Graph clustering on ego network	CBOW
TWE (Liu et al., 2015)	✗	LDA	Skip-gram
SenseEmbed (Iacobacci et al., 2015)	BabelNet	✗	CBOW
Chen et al. (2015)	WordNet	Context clustering	CNN
Jauhar et al. (2015)	WordNet	Expectation-Maximization	Skip-gram
Chen et al. (2014)	WordNet	✗	Skip-gram
Tian et al. (2014)	✗	Mixture of Gaussians	Skip-gram
Nieto Piña and Johansson (2015)	SALDO	✗	Skip-gram
MSSG (Neelakantan et al., 2014)	✗	✓	Skip-gram
SAMS (Cheng and Kartsaklis, 2015)	✗	✗	Recursive Neural Network
Li and Jurafsky (2015)	✗	Chinese Restaurant Process	CBOW-Skip-gram, SENNA
MSWE (Nguyen et al., 2017)	✗	LDA	Skip-gram
Guo et al. (2014)	✗	Affinity Propagation Algorithm	RNNLM
LSTMEmbed (Iacobacci and Navigli, 2019)	BabelNet	✗	LSTM
Kumar et al. (2019)	Knowledge Graph Embedding	✗	Framework consisting of different types of Encoders

of their approach.

2.7 Sense Embeddings

Another drawback of early word embeddings is they unite all the senses of a word into one representation. In reality, however, a word gets meaning in its use and can mean different things in varying contexts. For example, even though the words "hot" and "warm" are very similar when they are used to refer to temperature levels, they are not similar in the sentences "She took a sip of hot coffee" and "He received a warm welcome". In the transition period to contextual embeddings, different supervised and unsupervised solutions are proposed for having sense embeddings.

Schütze (1998) was the first work aimed at identifying senses in texts. He defines the problem of word sense discrimination as the decomposition of a word's occurrences into same sense groups. This definition is unsupervised in its nature. When the issue becomes labelling those sense groups, the task becomes a supervised one and is named as word sense disambiguation. The reader can refer to Navigli (2009) for a comprehensive survey on word sense disambiguation and Camacho-Collados and Pilehvar (2018) for an in-depth examination of sense embedding methods and their development.

Table 2.4 provides a classification of the studies that we analyze in this section. The classification dimensions include unsupervised/supervised, knowledge base, probabilistic approach, and NN model.

At the outset, unsupervised learning is used to discriminate the different senses of

a word.

Reisinger and Mooney (2010) propose a multi-prototype based word sense discovery approach. In their approach (***R&M***), a word's all occurrences are collected as a set of feature vectors and are clustered by a centroid-based clustering algorithm. The resultant clusters (fixed number) for each word are expected to capture meaningful variation in word usage rather than matching to traditional word senses. They define the similarity of words *A* and *B* as the "maximum cosine similarity between one of A's vectors and one of B's vectors" and provide experimental evidence on similarity judgments and near-synonym prediction. Moreover, variance in the prototype similarities is found to predict variation in human ratings.

Following Reisinger and Mooney (2010), Huang et al. (2012) also aim at creating multi-prototype word embeddings. They compute vectors using a feed-forward neural network architecture with one layer to produce single prototype word vectors and then perform spherical k-means to cluster them into multiple prototypes. They also introduce the idea of using global context where the vectors of words in a document are averaged to create a global semantic vector. The final score of embeddings is then calculated as the sum of scores of each word vector along with the global semantic vector.

The authors also argue that available test sets for similarity measurements are not sufficient for testing multi-prototype word embeddings because the scores of word pairs in those test sets are given in isolation, which lacks the contextual information for senses. Therefore, they introduce a new test set in which the word pairs are scored within a context by mechanical turkers, where context is usually a paragraph from Wikipedia that contains the given word. Finally, they show that their model is capable of outperforming the former models when such a test set is used, although its performance is similar to others in previous test sets.

Pelevina et al. (2016) aim at creating sense embeddings without using knowledge bases. Their model takes existing single-prototype word embeddings and transform them into multi-prototype sense embeddings by constructing an ego network and performing graph clustering over it. The senses of a word they learn do not have to correspond to the sense of that word in the dictionary. They evaluate their method on their crowd-sourced dataset.

Liu et al. (2015) propose three different methods to create topical embeddings (***TWE***). They create their topical embeddings without the use of any knowledge base but instead rely on LDA (Blei et al., 2003) to find the topics of each document the word occurs in. Topical embeddings they create are similar to sense embeddings with the only difference being that the number of topics may not correspond to the number of senses in the dictionary.

In their first model, named TWE-1, they learn word embeddings and topic embeddings separately and simultaneously with the skip-gram method by treating topic

embeddings as pseudo-words, which appear in all positions of words under this topic. Sense embeddings of a word w for topic t is then constructed by concatenating the word embedding w with the corresponding topic embedding t. Their second model TWE-2 treats word embeddings and topic embeddings as tuples and train them together. This method may lead to sparsity issues since some words on a specific topic may not be frequent. The last method they propose, TWE-3, also train word and topic embeddings together but this time the weights of embeddings are shared over all word-topic pairs. They show that the TWE-1 method gives the best results overall and the independence assumption between words and topics in the first model is given as the reason behind its performance.

Exploiting vast information in knowledge bases to learn sense representations has proved useful. Approaches that rely mainly on knowledge bases to compute sense embeddings include Iacobacci et al. (2015), Chen et al. (2015), Jauhar et al. (2015), and Chen et al. (2014).

Iacobacci et al. (2015) (***SenseEmbed***) use BabelNet (Navigli and Ponzetto, 2012) as a knowledge-base to retrieve word senses and to tag words with the correct sense. They train the sense-tagged corpora on the CBOW architecture and achieve state-of-the art results in various word similarity and relatedness datasets.

Chen et al. (2015) also use a knowledge base (WordNet) to solve the sense embedding problem. They use CNN to initialize sense-embeddings from the example sentences of synsets in WordNet. Then, they apply context clustering to create distributed representations of senses. The representation they obtain achieves promising results.

Jauhar et al. (2015) propose two models for learning sense-embeddings using ontological resources like WordNet (Miller, 1995). In their first model, they retrofit pre-trained embeddings by imposing two conditions on them: pulling together the words that are ontologically related (by using the graphs constructed from the relationships in WordNet) and leveraging the tension between sense-agnostic neighbours from the same graph. They implement the first method over Skip-gram (Mikolov et al., 2013) and Huang et al. (2012) and show that their method can improve the success of the previous methods. Their second method constructs embeddings from scratch by training them with an Expectation-Maximization (EM) objective function that pulls together ontologically-related words similar to the first model and finds the correct sense of the word from WordNet and creates a vector for each sense.

Chen et al. (2014) propose a unified model for word sense representation (WSR) and word sense disambiguation (WSD). The main idea behind this is that both models may benefit from each other. Their solution is composed of three steps: First, they initialize single-prototype word vectors using Skip-gram (Mikolov et al., 2013) and initialize the sense embeddings using the glosses in WordNet (Miller, 1995). They take the average of words in WordNet synset glosses to initialize the sense embeddings. Second, they

perform word sense disambiguation using some rules on the given word vectors and sense vectors, and finally using the disambiguated senses, they learn sense vectors by modifying the Skip-gram objective such that both context words and context words' senses must be optimized given the middle word in context.

Tian et al. (2014) propose a probabilistic approach to provide a solution to sense embeddings. They improve the Skip-gram algorithm by introducing the mixture of Gaussians idea to represent the given middle word in context in the objective function. Every Gaussian represents a specific sense and the mixture is their multi-prototype vector. The number of Gaussians, in other words, the number of senses is a hyperparameter of the model. They use Expectation-Maximization (EM) algorithm to solve the probabilistic model.

Nieto Piña and Johansson (2015) extend the Skip-gram (Mikolov et al., 2013) method to find sense representations of words. They get the number of senses from a knowledge base and for each word in the training corpus, they find the most probable sense by using likelihoods of context words. They only train the sense with the highest probability. They train their system on Swedish text and measure their success by comparing the senses to the ground-truth in the knowledge base (SALDO (Borin et al., 2013)).

Neelakantan et al. (2014) (***MSSG***) also aim at creating word vectors for each sense of a word. Different from most other models, they do it by introducing the sense prediction into the neural network and jointly performing sense vector calculation and word sense discrimination. Their first model relies on Skip-gram and induces senses by clustering context word representations around each word. Then, the word is assigned to the closest sense by calculating the distance to the sense-clusters' centres. Here the count of clusters is the same for all words and is a hyperparameter. Their second model is a non-parametric variant of the first one where a varying number of senses is learnt for each word. A new cluster (sense) for a word type is created with probability proportional to the distance of its context to the nearest cluster (sense). They show that their second method can outperform the first since it can learn the nature of senses better.

Cheng and Kartsaklis (2015) consider capturing syntactical information to better address senses. They use recursive neural networks on parsed sentences to learn sense embeddings. Each input is disambiguated to its sense by calculating the distance of average of the words' embeddings in the sentence to sense cluster means. They define two negative sampling methods to train the network. One negative example is created to swap the target word with a random word (as in Mikolov et al. (2013) and Gutmann and Hyvärinen (2012)), another negative sampling is done by changing the order of words in a sentence which further enforces the model (***SAMS***) to learn syntactic dependencies.

Li and Jurafsky (2015) decide the number of senses in an unsupervised fashion by using the Chinese Restaurant Process (CRP). They combine the CRP with neural network

training methods by deciding the sense of a word by looking at its context. They also compare sense-embedding methods with single-prototype models across various NLP tasks to see if they really are beneficial. They state that in some tasks (POS tagging, semantic relatedness, semantic relation identification) sense-embeddings outperform single-prototype methods, but they fail to improve their score on some other tasks (NER, sentiment analysis).

Instead of getting the number of senses from a knowledge base, Nguyen et al. (2017) (***MSWE***) use LDA (Blei et al., 2003) to find the word to topic and topic to document probability distributions. Here the number of topics is a parameter to the model. They train different weights for each sense of a word using two different optimization methods. The first model learns word vectors based on the most suitable topic. On the other hand, their second model considers all topics to learn them. They conclude that this second method can be considered as a generalization of the Skip-gram model (Mikolov et al., 2013) given the fact that it behaves as Skip-gram if the mixture weights are set to zero.

Guo et al. (2014) exploit bilingual resources to find sense embeddings, motivated by the idea that if a word in source language translates into multiple words in the target language that means different words in target language corresponds to a sense in the source language. For this purpose, they use Chinese to English translation data to induce senses in an unsupervised fashion. They represent the initial words with word embeddings from C&W (Collobert and Weston, 2008) and use affinity propagation algorithm to cluster the translated words into dynamic clusters which means that their method can learn different number of senses for each word. Then, they use the RNNLM model (Mikolov et al., 2010) to train the sense embeddings.

Iacobacci and Navigli (2019) propose an LSTM-based architecture (***LSTMEmbed***) to jointly learn word and sense embeddings. Input contexts are provided from semantically annotated data and one bidirectional LSTM processes the left context while another one handles the right one. As an extra layer, the concatenation of both outputs is linearly projected into a dense representation. Then, the optimization objective tries to maximize the similarity between the produced dense output and pre-trained word embeddings from SGNS. Consideration of these pre-trained word embeddings in the final phase increases the vocabulary use of the proposed system. Their experiments on word to sense similarity and word-based semantic evaluations prove the usefulness of their approach.

Kumar et al. (2019) propose a framework that combines context encoder with definition encoder to provide sense predictions for out-of-vocabulary words. In the case of rare and unseen words, most Word Sense Disambiguation (WSD) systems rely on the Most-Frequent-Sense (MFS) on the training set. In the part of definition encoder, sentence encoders along with knowledge graph embeddings are utilized. Here instead of using discrete labels for senses, the score for each sense in the inventory is calculated by the dot product of the sense embedding with the projected context-aware embedding.

Table 2.5: Morpheme Embedding Models.

Model	Year	Training Corpus	Knowledge-Base	NN Model	Dimension
Luong et al. (2013)	2013	Wiki	Morfessor	recNN	50
CLBL	2014	ACL MT	Morfessor	LBL	-
Qiu et al. (2014)	2014	Wiki	Morfessor,Root,Syllable	CBOW	200
Bian et al. (2014)	2014	Wiki	Morfessor, WordNet, Freebase, Longman Dict.	CBOW	600
CharWNN	2014	Wiki	-	CNN	100
KNET	2015	Wiki	Morfessor, Syllable	Skip-Gram	100
AutoExtend	2015	Google News	WordNet	Autoencoder	300
Morph-LBL	2015	TIGER	TIGER	LBL	200
Soricut and Och (2015)	2015	Wiki	-	Skip-Gram	500
C2W	2015	Wiki	-	biLSTM	50
Cotterell et al. (2016)	2016	Wiki	CELEX	GGM	100
Fasttext	2016	Wiki	-	Skip-Gram	300
char2vec	2016	text8 (wiki)	-	LSTM+Attn	256
Kim et al. (2016)	2016	ACL MT	-	CNN+LSTM	300-650
LMM	2018	Gigaword	Morfessor	CBOW	200

2.8 Morpheme Embeddings

The quest for morphological representations is a result of two important limitations of earlier word embedding models. The first point is, words are not the smallest units of meaning in languages, morphemes are. Even if a model does not see the word *unpleasant* in the training, it should be able to deduce that it is the negative form of *pleasant*. Word embedding methods that don't take morphological information into account can not produce any results in such a situation. The second limitation is the data scarcity problem of morphologically rich languages and agglutinative languages. Unlike English, morphologically rich languages have many more nouns and/or verb forms inflected by gender, case, or number, which may not exist in the training corpora. The same thing is also valid for agglutinative languages in which words can have many forms according to the suffix(es) they take. Therefore, models that take morphemes/lexemes into account is needed.

Researchers propose several ways to target morphological information, in order to obtain sub-word information for solving rare/unknown word problem of earlier word embedding methods and also to have better representations of words for morphologically rich languages. While some of the works are proposed to train embeddings directly from morphemes/lexemes, others adjust the representations of other word embedding models. The summary of these models and their properties can be seen in Table 2.5.

There are two main ways for **training morpheme embeddings from scratch:** While some methods (Luong et al. (2013), Botha and Blunsom (2014), Qiu et al. (2014), Bian et al. (2014), Cui et al. (2015), Cotterell and Schütze (2015), Xu et al. (2018), Soricut and Och (2015)) propose to use tools or special rules for dissecting a text to its morphemes, others (Bojanowski et al. (2016), Cao and Rei (2016), Ling et al. (2015), Dos Santos and Zadrozny (2014)) prefer to use characters or character n-grams as input to

learn morphemes along with their representations.

Luong et al. (2013)'s work is the first work that attempts to incorporate morphological information in word embeddings. They train morphological embeddings with recursive neural networks. They divide words into (prefix, stem, affix) tuples by using morfessor (Creutz and Lagus, 2007) and feed them to a recursive neural network. Word embeddings are then constructed by a word-based Neural Language Model (NLM). Instead of initializing the vectors with random numbers, they initialize them with pre-trained word embeddings from Collobert et al. (2011) and Huang et al. (2012) in order to focus on learning the morphemic semantics.

Similar to Luong et al. (2013), Botha and Blunsom (2014) (***CLBL***) also use morfessor (Creutz and Lagus, 2007) to find the morphemes of words in text and train both the target word and context words by first factoring them into their morphemes. They learn morphology-based word representations with an additive-LBL of their factor embeddings e.g. surface form, stem, affixes, etc.

Qiu et al. (2014) incorporate morphemes into CBOW (Mikolov et al., 2013) architecture: instead of predicting a word from the context words, they propose to use both morphemes and words, as an input and for prediction. They control the relative contributions of words and morphemes with two parameters that weigh the information to be extracted from each input. They use three different tools for extracting morphemes from corpus: *morfessor* (Creutz and Lagus, 2007), *root*, and *syllable* (Liang, 1983a).

Bian et al. (2014) investigate three different methods for finding better representations for words and morphemes: First by transforming CBOW (Mikolov et al., 2013) into a new basis by using morphemes (segmented by using morfessor Creutz and Lagus (2007)) instead of words. They later represent words as the aggregate of the morphemes they are composed of. Second, they provide additional information to their first model by feeding semantic and syntactic information vectors as inputs along with the morpheme vectors. As semantic and syntactic information, they use synsets, syllables, syntactical transformation, and antonym and synonyms from Freebase (Bollacker et al., 2008), WordNet (Miller, 1995), and Longman dictionaries[1]. Finally, they use syntactic knowledge (POS tagging vector) and semantic knowledge (entity vector and relation matrix) as auxiliary tasks, where they use syntactic/semantic information as outputs around the centre word to be predicted. Their relation matrix consists of relations such as *belong-to* and *is-a* relation. They examine the effects of both semantic and syntactic information compared to the baseline model (CBOW) and report the relative effects of each of them in various tasks.

Soricut and Och (2015) aim at improving word vectors and solving rare word problem by using morphology induction. In their method, they first extract candidate morphological rules. In this step, they find word pairs $(w1, w2)$ such that $w2$ is formed

[1]www.longmandictionariesonline.com

by substituting prefixes and suffixes up to 6 characters from $w1$ (i.e. $(bored, boring)$ is produced from the rule $(suffix : ed : ing)$). Later they form their rules from word pairs. After training their embeddings with the Skip-gram method (Mikolov et al., 2013), they keep the rule if the word pair $(w1, w2)$ is similar in embedding space, otherwise, the rule is removed from the candidate rule list. Thus, they use their morphological rules to obtain representations for rare words that may or may not be in the training set.

Cui et al. (2015) (***KNET***) use co-occurrence statistics to construct word embeddings with sub-word information. They leverage four different morphological information inspired by the advances in cognitive psychology: i) edit distance similarity ii) longest common sub-string similarity, iii) morpheme similarity (share roots, affixes, etc. by using morfessor (Creutz and Lagus, 2007)), and iv) syllable similarity (by using hyphenation tool (Liang, 1983b)). They combine the aforementioned morphological information into a relation matrix and construct morphological embeddings from it. On the other hand, they also create word embeddings by using the Skip-gram method (Mikolov et al., 2013). Combination of these two embeddings with weighted averaging is used to obtain the final word embeddings. Different from most other word embedding methods, authors do not change the digits in the text with zeros, instead, they change the digits with their text counterparts in order to reflect the information better.

Different from other morphology-based models, Cotterell and Schütze (2015) implement a semi-supervised approach (***MorphLBL***) where a partially morphologically tagged dataset (TIGER dataset of German newspaper (Brants et al., 2004)) is used. They augment the LBL model (Mnih and Hinton, 2007) to both predict word and morpheme together. They also introduce a new metric for measuring the success of morphological models called MorphDist.

Dos Santos and Zadrozny (2014), Ling et al. (2015), Bojanowski et al. (2016), and Cao and Rei (2016) come up with character-based solutions instead of using a tool/knowledge-base to find morphemes in sentences.

In their work (***CharWNN***), Dos Santos and Zadrozny (2014) use word embeddings together with character embeddings to compensate for the need for hand-crafted features in part-of-speech (POS) tagging, where the morphological structure of words plays a significant role. In their architecture, they use Skip-gram (Mikolov et al., 2013) for word embeddings and train their character embeddings from scratch.

The compositional model of Ling et al. (2015), called ***C2W***, takes characters of a word as input and uses bidirectional-LSTM (Graves and Schmidhuber, 2005) to construct word vectors by concatenating the last state of LSTM in each direction.

Bojanowski et al. (2016) propose a model, called ***Fasttext***, that takes character 3- to 6-grams of words and represent the words with bag of n-grams. i.e. for the word "where" the 3-grams are: (<wh, whe, her, ere, re>), where < and > are special characters for denoting the beginning and end of the word respectively. N-grams are then summed

to produce word embeddings. Thus, as the model shares representations across words, it is capable to have better representations for rare words. They perform extensive tests on morphologically rich languages to see how their model works and learns the subword information.

Cao and Rei (2016) aim at solving unsupervised morphology induction and learning word embeddings jointly by using bidirectional LSTMs (Hochreiter and Schmidhuber, 1997) with Bahdanau attention (Bahdanau et al., 2014) on characters. The output of the attention layer is fed to Skip-gram (Mikolov et al., 2013) algorithm to compute word representations. They prove that the attention layer learns how to split words into multiple morphemes by showing that their algorithm outperforms other morpheme induction methods although it is not only designed for solving that problem. They also show that since their method (***char2vec***) is focused on finding morpheme representations through characters, it is better at tasks that measure syntactic similarity. On the other hand, they argue that their method is worse at tasks that measure semantic similarity since characters do not convey any semantic information of words alone.

To address both syntactic and semantic features, Kim et al. (2016) use a mixture of character and word-level features. In their model, at the lowest level of the hierarchy, character-level features are processed by a CNN, after transferring these features over a highway network, high-level features are learned by the use of an LSTM. Thus, the resulting embeddings show good syntactic and semantic patterns. For instance, the closest words to the word *richard* are returned as *eduard, gerard, edward, and carl*, where all of them are person names and have syntactic similarity to the query word. Due to character-aware processing, their models can produce good representations for out-of-vocabulary words.

Xu et al. (2018) (***LMM***) also aim at enhancing word representations with morphological information. In incorporating morphological information, authors suggest using the latent meaning of morphemes instead of morphemes themselves. They state that although the words $incredible$ and $unbelievable$ have similar semantics, the methods based on morphemes cannot catch it. Instead, they use the latent meaning of morphemes that they extract from knowledge bases (i.e. in=not, un=not, ible=able, able=able, cred=believe, believ=believe). They use CBOW (Mikolov et al., 2013) as pre-trained word embeddings and show improvements using their method on them.

Among the models that adjust the pre-trained word embeddings, Rothe and Schütze (2015) take any word embeddings and transform them into embeddings for lexemes and synsets. In order to do that they use WordNet (Miller, 1995) synsets and lexemes although they note that their model (***AutoExtend***) can also get the information from other knowledge bases such as Freebase (Bollacker et al., 2008). They consider words and synsets as the sum of their respective lexemes and enforce three constraints on the system i) synset constraint ii) lexeme constraint and iii) WordNet constraint (due to the fact that

some synsets contain only a single word). They use an autoencoder where the result of the encoding corresponds to synset vectors, and the hidden layer in encoding and its counterpart in decoding corresponds to lexeme vectors. Two lexeme vectors are then averaged to produce the final lexeme embeddings.

On the other hand, Cotterell et al. (2016) use a Gaussian graphical model where words' embeddings are represented as the sum of their morphemes. Their system takes the output of other word embedding methods as input and converts them by learning their morpheme embeddings and calculating the word embeddings by summing them. They also note that with their method it is also possible to extrapolate the embeddings of OOV words since their morpheme embeddings can be calculated from the same morpheme in other words.

2.9 Contextual Representations

As is shown in the last sections, many methods have been proposed for solving the deficiencies of embedding methods. Each of them is specialized on a single problem such as sense representation, morpheme representation, etc., while none of them was able to combine different aspects into a single model, a single solution. It is the idea of *contextual representations* to provide a solution that covers each aspect successfully. The main idea behind contextual representations is that words should not have a single representation to be used in every context. Instead, a representation should be calculated separately for different contexts. Contextual representation methods calculate the embedding of a word from the surrounding words each time the word is seen, contrary to the earlier methods where each word is represented with a fixed vector of weights. This leads to an implicit solution to many problems such as sense representations, antonymy/synonymy, and hypernymy/hyponymy since now multi-sense words can have different representations according to their context. Furthermore, it has also been proposed to use characters as input which also incorporates the sub-word information into embeddings. Therefore, contextual representation models, described below, can incorporate different aspects together into a single model. Liu et al. (2020) examine contextual embeddings in detail by comparing their pre-training methods, objectives, and downstream learning methods.

In such a first attempt to create contextual representations, Melamud et al. (2016) develop a neural network architecture based on bidirectional-LSTMs to jointly learn context embeddings with target word embeddings. They feed words to a 2-layer bidirectional LSTM network in order to predict a target word in a sentence. They use sentences as context and feed the left side of the target word to the left-to-right (forward) biLSTM and

feed the right side of the target word to the right-to-left (backwards) biLSTM. To jointly learn context and target word embeddings, they use a Skip-gram objective function that is sampled on context-word occurrences. Furthermore, they show that this is equivalent to the factorization of a context-target word co-occurrence matrix. Although previous word embedding models create both context and target word embeddings, they only use target-target similarity as representations and ignore the context embeddings. In this work, authors also use context-context and context-target to show that contextual embeddings can improve the performance of NLP systems significantly. They also show that since bidirectional LSTM structures can learn long-term contextual dependencies, their model, ***context2vec***, is able to differentiate polysemous words with a high success rate.

CoVe (McCann et al., 2017) uses Glove (Pennington et al., 2014) as the initial word embeddings and feeds them to a machine translation architecture to learn contextual representations. The authors argue that pre-training the contextual representations on machine learning tasks, where there are vast amounts of data, can lead to better contextual representations to use as a transfer learning to other downstream tasks. They concatenate the output of the encoder of a machine translation model (as contextual embeddings) with GloVe embeddings to construct their final word representations.

Using language modelling and learning word representations as a pre-training objective then fine-tuning the architecture to downstream tasks is first proposed by Dai and Le (2015) and Howard and Ruder (2018). While Dai and Le (2015) propose to use RNNs and autoencoders to tackle the issue, ***ULMFiT*** (Howard and Ruder, 2018) introduces novel fine-tuning ideas such as discriminative fine-tuning, slanted triangular learning rates, and gradual unfreezing to their LSTM model, inspired from the advances in transfer learning in computer vision. After the success shown by these models, the aim is shifted from creating word representations to using their system as pre-trained models and then fine-tuning a classifier on top to perform on downstream tasks.

ELMO (Peters et al., 2018) improves on the character-aware neural language model by Kim et al. (2016). The architecture takes characters as input to a CNN network from where it is fed to a 2-layer bidirectional-LSTM network to predict a target word. They show that this architecture can learn various aspects of words such as semantic, syntactic, and sub-word information. First, they show that, since model takes characters as inputs, it is able to learn sub-word information even for the unseen words. Second, they show that while the first layer of biLSTM better captures the syntactic similarity of words, the second layer better captures the semantics. Therefore, they propose to use the different layers of the model to create word representations. They also propose to use a weighted averaging method for combining the different layers. They show that including ELMO representations can improve many state-of-the-art models in various NLP tasks.

Instead of using words as input, ***Flair*** (Akbik et al., 2018) uses a character-level language model to learn contextual word representations. Different from ELMO (Peters

et al., 2018) where character level inputs are later converted into word features, in this work authors propose to use characters only. They feed the characters of an input string to a single layer LSTM network and try to predict the next character. They, later, form the word representation by concatenating the backwards LSTM output from the beginning of the word with the forward LSTM output from the end of the word. They also try concatenating other pre-trained word vectors with their contextual representations in downstream tasks and show that this can improve the results.

BERT (Devlin et al., 2019) uses bidirectional transformer (Vaswani et al., 2017) architecture to learn contextual word representations. Different from the earlier approaches (ELMO (Peters et al., 2018), Melamud et al. (2016)) BERT is bidirectional. Although ELMO also considers both sides of a target word, it considers them separately as the left-side and right-side. Instead, BERT spans the entire sentence with both right-to-left and left-to-right transformers. To be able to do so, without also spanning the target word, they mask the target word. Therefore, they call this model, a masked language model (MLM).

In addition to the token (word) embeddings, they also use segment (sentence) embeddings and position embeddings (words' position in segments) as input which enables BERT to consider multiple sentences as context and to represent inter-sentence relations. Giving multiple sentences as input helps BERT to be integrated into most downstream tasks that require inter-sentence connection such as Question Answering (QA) and Natural Language Inference (NLI) easily without requiring any other architecture. For further details, the reader can refer to the work of Rogers et al. (2020), which provides an in-depth survey on how exactly BERT works and what kind of information it captures during training and fine-tuning.

XLNet (Yang et al., 2019) is an autoregressive method that combines the advantages of two language modelling methods: Autoregressive models (i.e. transformer-XL (Dai et al., 2019)) and autoencoder models (i.e. BERT (Devlin et al., 2019)). Specifically, It takes into account both sides of the target word by employing a permutation language modelling object without masking any words like BERT. This allows their model to capture also the relation between the masked word and the context words, unlike BERT.

ALBERT (Lan et al., 2020) aims at lowering the memory consumption and training times of BERT (Devlin et al., 2019). To accomplish this, they perform two changes on the original BERT model: They factorize the embeddings into two matrices to be able to use smaller dimensions and they apply weight sharing to decrease the number of parameters. They state that the weight sharing also allows the model to generalize better. They show that although they can obtain state-of-the-art results over BERT with fewer parameters, ALBERT requires longer time to train than BERT.

RoBERTa (Liu et al., 2019) revises the pre-training design choices of BERT (Devlin et al., 2019) by trying alternatives in a controlled way. Specifically, dynamic

masking for the Masked Language Model (MLM), input format of full sentences from a single document with the Next Sentence Prediction (NSP) loss removed, and byte-level Byte Pair Encoding (BPE) vocabulary give better performance. Moreover, they extend the training set size and the size of mini-batches in training. As a result, RoBERTa (Liu et al., 2019) achieves state-of-the-art results in GLUE, RACE, and SQuAD benchmarks.

In their work, called ***ERNIE***, Sun et al. (2019) improve on BERT by introducing two knowledge masking strategies into their masked language modelling. In addition to masking out random words in the sentence, they also mask phrases and named entities in order to incorporate real-world knowledge into language modelling/representation. In their successive work, ERNIE 2.0 (Sun et al., 2020), they implement continual multi-task learning. Including the one in ERNIE, they define seven pre-training tasks that are categorized into word-aware, structure-aware, and semantic-aware pre-training tasks, where they aim to capture lexical, syntactic, and semantic relations respectively.

GPT and its variants rely on a meta-learner idea by using a conditional language model in diverse NLP tasks. This conditional language model predicts the next word conditioned both on an unsupervised pre-trained language model and the previous set of words in context. In GPT-3, (Brown et al., 2020) pre-train a 175 billion parameter transformer-based language model on a sufficiently large and diverse corpus and tests its performance in zero-shot, one-shot, and few-shot settings. Their learning curves for these three settings show that larger a model is better in learning a task from contextual information. Authors apply task-specific input transformations e.g. delimiting context and question from the answer in reading comprehension, to test the model's performance in different NLP tasks. Their few-shot results prove the effectiveness of their approach by outperforming state-of-the-art on LAMBADA language modelling dataset (Paperno et al., 2016), TriviaQA closed book open domain question answering dataset (Joshi et al., 2017), and PhysicalQA (PIQA) common-sense reasoning dataset (Bisk et al., 2019).

2.10 Multi-Modal Word Embeddings

Initially, attempts to bring together textual and image features focused on using one type of information to enhance the results of the other. Instead of training a joint model, these works use the features of a trained model in one modality as target values or additional feature vectors in other modality to enhance the latter's performance. This method is called transfer learning.

One of the first such attempts at transfer learning was zero-shot learning of image classification methods. Zero-shot learning, as its name suggest, aim at successfully

predicting samples that model was not trained with. In order words, the models that can predict unseen samples is called zero-shot learning models. Earlier models of image classifiers (i.e. Deng et al. (2009), Szegedy et al. (2015), He et al. (2016)) were subject to two main drawbacks: Inability to predict unknown classes and failure to identify the relationship between different classes. Most of the image models are trained on the ImageNet image classification dataset (Deng et al., 2009) with 1000 labels where each label is considered distinct and independent from the others. Any model that is trained on these classes was unable to make any predictions when a new class is given to the model. Considering the fact that, there are new classes emerging in the world every day, such as new car brands/models, discoveries, etc., training these models from scratch every time a new class label is introduced, is very time consuming and not a feasible solution since it requires a lot of training time. In addition to that, there may not be enough training data for those emerging classes either. Therefore, there was a need for a model that can adjust to the unseen classes without requiring re-training the neural network all over again. The second problem of image models was the independence assumption among target classes. Neural image models such as AlexNet (Krizhevsky et al., 2012), GoogleNET (Szegedy et al., 2015) and Resnet (He et al., 2016), all use the softmax classification layer in order to make predictions. This method draws out the difficulty in differentiating a class label from another. However, nearly all classes in the imageNet dataset have a hierarchical structure: Some classes in the dataset are more similar than they are to other classes. For example, it is easier for any model (or any human annotator) to be confused about the classification of an image between the classes *"Australian terrier"* and *"Airedale terrier"* than to be confused between the classes *"Australian terrier"* and *"container ship"*. Even if a model makes a mistake in identifying the correct class label, it is more realistic and more acceptable to make that mistake with a label very similar to the ground-truth label rather than an irrelevant one.

2.10.1 Zero-Shot Learning

To solve the aforementioned problems, zero-shot learning methods propose to use word embeddings as target values instead of using the softmax layer and one-hot target vectors. This way the classifier is not limited to the number of classes in the training set (which is 1000), instead it is limited by the size of the vocabulary of word embeddings (which are in terms of millions). Furthermore, even if the model makes a mistake in the prediction, it picks a similar class label to the ground-truth which makes the errors more realistic and acceptable. In addition to the many benefits mentioned above, these models

also produce multi-modal embeddings as a side effect.

To our knowledge, the first neural model to leverage the textual information for zero-shot learning was Weston et al. (2010). They used a linear mapping from visual features into a feature space where both image features and annotations are represented.

Rohrbach et al. (2011) examine the effectiveness of various methodologies of knowledge transfer (KT) from text to image models: Hierarchy-based KT, attribute-based KT, and finally direct similarity-based KT.

In their zero-shot learning model, Socher et al. (2013) use the method of Coates and Ng (2011) to extract the set of features from images that are fed to a two-layer feed-forward neural network for mapping them to the corresponding vectors in embedding space. Huang et al. (2012) is used to create word embeddings. The authors also use novelty detection algorithms to decide whether a given sample is unknown or it belongs to an already trained class label.

DeViSE (Frome et al., 2013) trains separately an image model (AlexNet (Krizhevsky et al., 2012)) and a language model (word2vec (Mikolov et al., 2013)). Then, the trained word embeddings from the language model are used as target vectors for image classification tasks.

Instead of training the image models by replacing the softmax layer with word embeddings, ***ConSE*** (Norouzi et al., 2014) transforms the output of the softmax layer to an embedding space. By using such transformation, they manage to build a zero-shot learning system with the existing pre-trained models without requiring any further training.

Kodirov et al. (2017), on the other hand, states that naively replacing the target one-hot vectors with word embeddings leads to domain shift problems. Although the model can predict the labels that it does not train with in the first place, it does not preserve the information coming from the textual embeddings after training. By introducing their autoencoder approach, which tries to reconstruct the image input with the same embedding (word2vec embeddings Mikolov et al. (2013) are used in their work) that it tries to predict, they aim at overcoming the domain shift problem.

Xian et al. (2018) target the data imbalance problem between seen and unseen classes. They propose a generative adversarial network model to circumvent the issue and obtain a significant performance increase in many zero-shot learning tasks.

Similar to Norouzi et al. (2014), Changpinyo et al. (2016) also propose to transform the model space into a semantic space. To do so, they use manifold learning on various embedding models and image models.

Romera-Paredes and Torr (2015) propose a linear mapping approach for training the image models with semantic labels. Different from Norouzi et al. (2014) and Changpinyo et al. (2016), they use attribute signatures of images instead of using the outputs of image models such AlexNET (Krizhevsky et al., 2012), Resnet (He et al., 2016), etc.

Xie et al. (2019) were the first to implement an attention mechanism in a zero-shot

learning problem. For further improving the results of zero-shot learning systems, they use an attentive regional embedding network with Resnet (He et al., 2016). Their network can learn how to attend to image regions depending on the semantic embeddings used in the sample.

As an alternative way of accomplishing the task of zero-shot learning, researchers proposed to use generative adversarial networks (GAN) to create samples for unseen classes. With this approach, for each unknown class, a pre-trained GAN model is run to create samples from its embedding. Later, these synthesized samples are used to train the model. This scheme allows the models to implicitly learn the textual features and use traditional classification layers such as softmax.

In such a work, Ma et al. (2020) used a Wasserstein GAN to create synthesized images from semantic embeddings to enhance the classification of unseen classes. Although they mostly experimented with datasets which has a lower number of classes than imageNET (Deng et al., 2009), they report a significant increase in performance.

Zhu et al. (2018) also make use of a GAN to create synthesized images, but instead of using embeddings to create images, they used noisy textual descriptions (i.e. Wikipedia articles).

These models and many more that followed them in their footsteps were able to generalize better to unknown labels, making the models more successful for downstream tasks. For further information on zero-shot learning, the reader can refer to the surveys of Xian et al. (2019) and Wang et al. (2019).

2.10.2 Multi-Modal Representations and Language Models

Various studies examined the use of visual information for training language models. While some of those studies focused on producing **better representations** (Andrews et al. (2009), Bruni et al. (2014a), Bruni et al. (2012), Kiros et al. (2014a), Liu et al. (2017), Hill and Korhonen (2014), Ororbia et al. (2019), Kiros et al. (2014b)), most of these models produce multi-modal embeddings as a side-product of a multi-modal task. These tasks include **image retrieval with text/captioning** (Karpathy et al. (2014), Karpathy and Fei-Fei (2017), Wang et al. (2016)), **image-text alignment** (Lee et al. (2018), Socher and Fei-Fei (2010)), **image segmentation using a target text** (Yu et al. (2018)), **visual question answering** (Anderson et al. (2018), Agrawal et al. (2018), Gao et al. (2019)), **visual common-sense reasoning** (Zellers et al. (2019)), and **image captioning** (Kiros et al. (2014b)). Some other studies also contributed to the field of multi-modal language modelling by encompassing many of these models similar to contextual embed-

ding (Lu et al. (2019)) or by enhancing the existing models (Shi et al. (2018)). As the field is relatively new, most of these works focus on the fusion of modalities more than the individual models.

One of the first models to experiment with multi-modal representations is Andrews et al. (2009). In their work, they draw attention to advances in cognitive science where it is shown that language acquisition in children mostly relies on experiential data. They use LDA (Blei et al., 2003) on human generated data (experiential) and text (distributional) to calculate multi-modal representation of words. They create three models: one model from experiential data, one model from text and finally a combined LDA model that is created from the combined data.

Similar to Andrews et al. (2009), the ***mixLDA*** model of Feng and Lapata (2010) also ground their reasoning to advances in cognitive science which states that human language perception relies significantly on experiential data. They use a modified version of LDA to create their multi-modal representations.

Socher and Fei-Fei (2010) proposed the use of canonical correlation analysis (CCA) for image-text alignment. They were able to form a joint latent meaning space of vision and text for words.

Leong and Mihalcea (2011) use non-neural methods to find representations for text and image and combine them by summing their relatedness ratings. They show that information from images improves the models' success in word relatedness tasks such as MC (Miller and Charles, 1991), RG (Rubenstein and Goodenough, 1965) and WS (Finkelstein et al., 2001).

Kiros et al. (2014a) use a language model based on LBL (Mnih and Hinton, 2007). They learn the features of images through a CNN and feed them to their ***MLBL*** language model along with the textual inputs.

Hill and Korhonen (2014) extend the word2vec model (Mikolov et al., 2013) to incorporate the experiential information. They used the ESP game dataset (von Ahn and Dabbish, 2004) and CSLB dataset (Devereux et al., 2014), where the objects in the image are given for each sample, for obtaining visual features. While training the word2vec model, if one of the words in the image dataset is encountered, the model is also run with the pseudo-sentence created by concatenating object words in the image.

Instead of creating a joint embedding space of image and text, Karpathy et al. (2014) and Karpathy and Fei-Fei (2017) create image and text representations separately and use an alignment objective on the pairs of visual and textual features. Also, different from many approaches mentioned here, they use dependency relations as input to the text encoder instead of words in isolation.

Bruni et al. (2014a) and Bruni et al. (2012) combine the Scale-Invariant Feature Transform (SIFT) feature vectors (Lowe, 1999) with word embeddings through weighted averaging to form multi-modal embeddings. They show that their model performs worse

than textual models in semantic relatedness tasks but outperform them in others.

Instead of fusing the multi-modal information together, ***SC-NLM*** (Kiros et al., 2014b) learns separate embeddings and try to project them into a single joint embedding space by pulling the image-text pairs together using pairwise ranking loss. In their work, they also show that algebraic operations on multi-modal embeddings also work by showing that *image of a blue car* - "blue" + "red" is near to the images of red cars.

Wang et al. (2016) use Hybrid Gaussian-Laplacian mixture model (HGLMM) (Klein et al., 2015) and VGG-19 (Simonyan and Zisserman, 2015) to obtain features and feed them through two feed-forward layers. They use the inner product of the modalities in order to fuse them.

RRFNet (Liu et al., 2017) introduces the recurrent residual fusion (RRF) blocks to further enhance and bridge the gap between textual and vision features they obtained with Hybrid Gaussian-Laplacian mixture model (HGLMM) (Klein et al., 2015) and Resnet-152 (He et al., 2016) respectively. To combine the enhanced outputs of RRFs, they use the inner product and calculate the loss with the bidirectional rank loss.

The method of Eisenschtat and Wolf (2017) differs from the other works in this field in terms of the training scheme. Instead of training visual and textual inputs together with a common classification task, they use a 2-way neural network architecture with three feed-forward layers where each network receives the input in one modality and try to reconstruct the input of the other modality. They use canonical correlation analysis (CCA) on the middle layers with euclidean loss to train the network.

Shi et al. (2018) state that the multi-modal language models are vulnerable to adversarial attacks. To overcome this issue, they train their model, ***VSE-C***, with adversarial examples that they created from the COCO dataset (Sharma et al., 2018). They fuse the word embeddings with the image features from Resnet-152 (He et al., 2016) using the embedding interaction method (Gong et al., 2017).

Collell Talleda et al. (2017) learn a mapping function that maps the textual embeddings to the output of a CNN, therefore learning to "imagine". Later, they concatenate the textual embeddings with their corresponding mapped vector during testing to form multi-modal embeddings.

MAttNet (Yu et al., 2018) aims at image segmentation with textual information. Their model finds the segments in images that are described by the textual information such as: "person on the left" or "the women with the short red hair". They use a bidirectional-LSTM (Hochreiter and Schmidhuber, 1997) and experimented with both Resnet (He et al., 2016) and faster-RCNN (Ren et al., 2015) for each modality respectively.

Anderson et al. (2018) propose to combine the image and textual features through bottom-up and top-down attention mechanisms. They use an LSTM to compute a representation for text and a combination of Resnet (He et al., 2016) and faster-RCNN (Ren et al., 2015) to find image features. Later, they perform a weighted sum over the image

features with the output of attention mechanism (either bottom-up or top-down) that uses both image and text representations. Finally, they combine these weighted features of images with textual features to feed to the final feed-forward layers and a classification layer.

The ***SCAN*** (Lee et al., 2018) model uses the bottom-up attention mechanism for image-text alignment, where the corresponding object is found in the image for each word in the sentence. They compute the representation for images with a modified version of the combination of faster-RCNN (Ren et al., 2015) and Resnet (He et al., 2016) and find textual representations through a bidirectional GRU (Cho et al., 2014). They combine the two modalities through a Stacked Cross Attention Network where the image patch is compared to the attended sentence representation.

Lu et al. (2019) propose a concurrent pre-training scheme of image and text models, called ***VilBERT***. To accomplish that they use two BERT (Devlin et al., 2019) models combined with the co-attention layers proposed by Ren et al. (2015). Each BERT model is responsible for a different modality. One BERT model takes the text input while the other processes the image features obtained with the object detection model Faster-RCNN (Ren et al., 2015). Both model outputs are given to the multi-attention heads of the other through the co-attention layers in order to form a joint feature space. Their training objectives are followed from the original BERT model: masked multi-modal modelling and multi-modal alignment prediction.

GVQA (Agrawal et al., 2018) has a similar approach to VilBERT by leveraging an LSTM model (Hochreiter and Schmidhuber, 1997) for text input using GloVe embeddings (Pennington et al., 2014) and VGG model (Simonyan and Zisserman, 2015) combined with Stacked Attention Networks (SAN) (Yang et al., 2016) for image input. The major difference between the two models is that the GVQA directly uses a pre-trained image model without extra training and combine the models at the test time while VilBERT also performs the pre-training on the combined model with the Microsoft COCO dataset (Sharma et al., 2018). Although this extra training gives VilBERT an advantage over GVQA, it might also introduce a training bias, since the VQA dataset (Antol et al., 2015) also uses the COCO images.

In addition to their proposed model, the authors also state that the original VQA dataset (Antol et al., 2015) contains bias and relies on language priors where questions such as "what is the colour of" always leads to white/no. This causes a poor generalization on test sets. To overcome this issue, they use different splits in training and test sets and report their results on this modified dataset.

The ***DFAF*** (Gao et al., 2019) model uses faster-RCNN (Ren et al., 2015) and GRU (Cho et al., 2014) network with GloVe embeddings (Pennington et al., 2014) for encoding image and textual inputs respectively. After computing the features, they are passed through inter and intra modality attention modules which attend over each modality

individually and together to learn cross-modal features. They state that using attentions between and across the modalities can lead the system to capture high-level interactions between the language and vision domains.

The ***R2C*** model (Zellers et al., 2019) targets commonsense reasoning of movie scenes. In their dataset, images from the movie scenes are paired with commonsense reasoning questions such as "Why is [person1] pointing at [person2]?". The aim of the task is to find the corresponding answer among given choices. Their model for solving this task consists of three parts: grounding, contextualizing, and reasoning. In the grounding part, a joint representation of image and text is found through bidirectional-LSTMs (Hochreiter and Schmidhuber, 1997) and a CNN. The contextualization part involves an attention mechanism between query and response over the learned representations. Finally, in the last part, the reasoning, another bidirectional-LSTM is used to generate reasoning.

Anastasopoulos et al. (2019) experiment with various strategies for combining the experiential information and textual information: combining them before feeding them to the model (early fusion), combining them between the layers of the model (middle fusion), and combining them after all the layers (late fusion). They also tried combining them as a linear combination of the outputs in the classification layer. They show that the middle fusion works the best. The major difference between their model and the others is that they use videos instead of images for experiential information. As a language model, they use a two-layered LSTM (Hochreiter and Schmidhuber, 1997) with wordpiece embeddings and video embeddings. Although they did not test their model on downstream tasks, they show that their model works with language model objectives.

Unified-VSE (Wu et al., 2019) also aims at forming a joint space of visual and semantic embeddings, but the difference of their methods from the others is that they form representations for different semantic components: objects, attributes, relations, and scenes. They use the GloVe (Pennington et al., 2014) embeddings with a uni-directional GRU (Cho et al., 2014) for textual parts and the Resnet-152 model (He et al., 2016) for image parts. They use alignment losses for each semantic component and negative sampling for training their joint model.

Ororbia et al. (2019) integrated the visual features they obtain with the inception model (Szegedy et al., 2016) in various language models (RNN, LSTM (Hochreiter and Schmidhuber, 1997) and GRU (Cho et al., 2014)). It is accomplished by feeding the image features into every time-step of RNN/LSTM/GRU. Their reasoning was similar to ours: Language is inseparable from the physical or social context.

Among these works, the closest to our study are VilBERT (Lu et al., 2019), and Hill and Korhonen (2014). VilBERT is the only work to propose the use of additional multi-modal pre-training similar to ours, but the difference between our model and theirs is we take advantage of the concreteness information similar to curriculum learning (Bengio et al., 2009), while they merely train with multi-modal data without regard for any inherent

information. Hill and Korhonen (2014) on the other hand, do not perform extra pre-training steps although they perform multi-modal training during the first pre-training phase. Finally, the essential difference is, even though their model can differentiate the concreteness levels of the words explicitly (they only train their model on images if one of the words appears in images as objects), they compute their experiential features from the text too, where the image and textual features do not align. As experiential features, they do not use images themselves but use the textual embeddings of the objects that exist in a particular image as if it is a sentence. Consequently, this leads to unaligned image-text pairs during training.

2.11 Curriculum Learning

Many deep learning methods use randomly ordered samples during training which contradicts how the humans learn the concepts: We start learning with easy concepts and gradually increase the difficulty of the task as we get better. Curriculum learning (CL), as a study, aims at creating a meaningfully ordered training set in terms of difficulty, mimicking the human learning process.

Although curriculum learning has risen popularity in the last few years, similar to neural network methods, it also has a long history of research dating back to the work of Newport (1990) and Elman (1993).

In his "less is more" hypothesis, Newport (1990) provides empirical evidence that early learners of languages are significantly more successful than the late learners. It is shown that this is mainly due to starting from "less" towards "more" in terms of language knowledge that is already acquired.

In his work, Elman (1993) shows that "starting small", both in terms of difficulty of the training test and in terms of the the size of the network used, benefits language learning process significantly. It is argued that this benefit comes mainly from the fact that starting small reduces the number of error minima in the system therefore prevents the system from falling into a local minima where it is very hard to escape from in later stages of the training.

Bengio et al. (2009) was the work that popularized the idea of using a curriculum learning for training deep neural networks. In their paper, they show that the curriculum learning can help neural networks to converge faster and find a better minima in the case of non-convex optimization.

Depending on how they classify a sample as easy or hard, the curriculum learning literature is divided into various sub-fields, described below (For further details on

curriculum learning refer to the surveys of Soviany et al. (2021) and Wang et al. (2021)).

Traditional curriculum learning, also called as vanilla CL or easy-to-hard CL, uses prior knowledge on the dataset to divide the samples into easy and hard. Mostly, this separation is provided by the researchers heuristically, based on predefined difficulty criteria on a sub-field.

Spitkovsky et al. (2009) use an easy-to-hard curriculum training for unsupervised dependency parsing. Their easy samples are formed from short sentences in which it is easier to determine the dependency (trivial case being one word sentences where the dependency is on the word itself), to long sentences in which the complexity of determining the dependency is increased (sentence of up to 45 words where there might be dozens of words in between the dependents).

Caubrière et al. (2019) apply easy-to-hard CL to spoken language understanding where the samples are ordered from the most generic concepts to more specific ones. They define the generic as the level of information conveyed in the the training samples: most generic samples are just plain words while the less generic ones also contain additional information such as named entity tags.

Similar to Caubrière et al. (2019), Shi et al. (2013) also order the training samples from generic to specific. In their case, their language model is trained with the texts of more generic topics first, then the specific contexts specialized on various topics are used to conclude the training.

Zaremba and Sutskever (2014) use easy-to-hard CL training for learning to understand and execute computer programs with language models. Different from other vanilla CL work, they also include some difficult examples early on in the training and increase their ratio gradually throughout the training.

Instead of ordering samples, Kim et al. (2019) order the tasks from easy to hard in order to train their multi-modal question answering model. Their model is composed of three tasks: The easiest task, modality alignment is given priority at the beginning of the training. Then, the temporal localization task is focused on. Finally, the question answering task is set to higher priority for the training.

Instead of using prior knowledge, **Self-Paced Learning (SPL)** suggests using the models' own feedback as a measure of difficulty. Samples are classified as easy or hard on the fly during training, using various metrics such as classification loss, confidence intervals of prediction, etc.

Kumar et al. (2010) classify samples as easy if it is easy to predict the correct output. They compute this with the degree of certainty of the model predictions.

Lee and Grauman (2011) determine the easiness of objects in images dynamically with two criteria: 1) objectness: likelihood of object being in any generic category, 2) context-awareness: likelihood of its surrounding objects are of familiar categories. They start with a set of generic categories such as grass, sky, road, etc. and add the easiest

unknown object category to it one by one throughout the training.

Xu et al. (2020) propose a SPL model for fine-tuning the language models on natural language understanding. They use the success of their initial model as a deciding factor for the difficulty of the samples and built their training set accordingly.

Xu et al. (2018) propose a self-paced learning approach to multi-modal image classification. Their model feeds the classification losses for each modality into curriculum learning module by using the modality weights. Curriculum learning model is, then used for updating the modality weights. With this method, samples are fed into the training from easy to hard gradually until the model converges.

Self-Paced Curriculum Learning (SPCL) is a combination of self-paced learning and vanilla CL. It aims at combining the advantages of both by taking into account the prior knowledge (as in vanilla CL) and the feedback from the learner (as in SPL).

Jiang et al. (2015) introduce the idea of self-paced curriculum learning, where the learning scheme in curriculum learning is merged with the dynamic setting of self-paced learning. SPCL manages to create a "student-teacher collaborative learning" environment by both taking prior knowledge as in curriculum learning and dynamically arranging training samples in terms of difficulty. A regularization term is used to determine the difficulty of samples during training based on their losses.

All of the aforementioned CL training methods divide the training samples into discrete categories depending on the prior knowledge, knowledge obtained during training, or both. **Progressive CL**, on the other hand, does not use any discrete categories of training samples. Instead, it aims at gradually increasing the difficulty of training samples continuously throughout the training.

Morerio et al. (2017) propose to increase dropout rate for image inputs in order to change difficulty at training. They state that the negative co-adaptations in neural network occur during the later stages in training, therefore using a constant drop-rate is not optimal. They show that using dynamic dropout-rate in a curriculum learning fashion provides better results in training.

Braun et al. (2017) implement a progressive CL method on automatic speech recognition by gradually increasing the noise from 0dB to 50dB as the training goes on.

Teacher-Student CL uses two models during training in order to create a curriculum: A teacher network that decides which samples are easy and which ones are hard, and a student network that learns the task at hand. While some studies use a static teacher network with unaltered weights during training, some others train the teacher network as well, alongside the student network.

Kim and Choi (2018) train two networks jointly: a student network that is trained to determine the significance of the samples into easy and hard cases and a main network that learns the task at hand. The significance of samples is computed using the losses of both networks at each iteration in training.

Hacohen and Weinshall (2019) use a pre-trained teacher network (Inception Szegedy et al. (2016)) to classify the samples into easy and hard cases based on the confidence level of predictions. Then, the student network is trained with the samples with gradually increasing difficulty.

Gong et al. (2016) and Gong (2017) propose a teacher-student CL algorithm for multi-modal classification tasks. It is argued in the paper that selecting easy samples over all modalities with a common curriculum can be misleading since it does not take into account the individuality of each modality. Therefore, they propose to use multiple teacher models that work on an individual model, then, their outputs are used to decide the easy samples for the student model. As a result, this "soft fusion" technique is able to consider the individual characteristics of each modality while still being subject to common curriculum criteria.

The final way of meaningfully building a curriculum is the **Active Learning**, which orders the training samples using uncertainty rather than assessing the difficulty.

Chang et al. (2017) propose a system where the uncertainty in the prediction is used as the method of distinction among samples instead of difficulty. Priority is given to samples with low prediction variance, and later the model uses more the samples with high prediction variance (samples that are predicted as different classes in different runs).

Tang and Huang (2019) combine the Active learning with self-paced learning to benefit from the advantages of both methods. Their self-paced active learning method simultaneously considers both the easiness of examples and their informativeness through the use of uncertainty. They argue that although the active learning methods take into consideration the informative and representative samples, they might select over-complex samples in early stages of training. Therefore, the samples cannot be utilized fully on the model at hand.

Lotfian and Busso (2019) implement an active learning framework for multi-modal curriculum learning for speech emotion recognition. They use the disagreement between the human annotators as the source of uncertainty, and use it to order the training samples from the easier, unambiguous ones to harder and ambiguous cases.

There have been some studies among the aforementioned work that focused on implementing curriculum learning on language models (Bengio et al., 2009; Shi et al., 2013; Xu et al., 2020; Zaremba and Sutskever, 2014) and some studies that focused on using curriculum learning on multi-modal tasks (Gong et al., 2016; Gong, 2017; Kim et al., 2019; Xu et al., 2018; Lotfian and Busso, 2019). But, to our knowledge, this is the first work that brings all these aspects together into a single model and a single training scheme (Sezerer and Tekir, 2021). The training methodology in this work falls into easy-to-hard curriculum learning category, since it uses prior knowledge to order the samples.

2.12 Evaluation of Embedding Models

Due to the popularity of the field, many datasets are proposed and tested upon. In this section, we report the structure of the datasets and the performance of the aforementioned word embedding models on them.

2.12.1 Datasets

Depending on their aim, datasets produced to measure the success of embedding models can be divided into four categories: Similarity tasks, Analogy task, Synonym selection tasks, and Downstream tasks.

2.12.1.1 Similarity Tasks

These datasets provide pairs of words whose similarity is rated by human judgements. They all use Spearman's rank correlation (ρ) with average human judgement to measure the performance and quality of embeddings.

- WordSim-353 (WS-353): Finkelstein et al. (2001) produced a corpus that contains human judgements, rated from 1 to 10, on 353 pairs of words.
- SCWS: Huang et al. (2012) introduced this dataset in which the word pairs are scored by mechanical turkers, within a context, which is usually a paragraph from Wikipedia that contains the given word. The reason for introducing such a dataset is that the available test sets for similarity measurements are not sufficient for testing multi prototype word embeddings because the scores of word pairs in those test sets are given in isolation, which lacks the contextual information for senses.
- RG-65: This dataset, developed by Rubenstein and Goodenough (1965), is composed of 65 noun pairs whose similarity is rated by human annotators.
- MC-30: The dataset (Miller and Charles, 1991) contains 30 pairs of word swhose similarity is rated by human annotators.
- MEN: It (Bruni et al., 2014b) contains 3000 pairs of words together with human assigned similarity score obtained from Amazon Mechanical Turk.

- YP-130: Similar to previous test sets, YP-130 (Yang and Powers, 2005) also contains human assigned similarity scores to 130 word pairs.
- RW: Unlike previous word similarity datasets, RW (Luong et al., 2013) consists of 2034 pairs of rare words which are not frequently seen in texts. The motivation behind this dataset is to provide sufficient number of complex and rare words to test the expressiveness of morphological models since previous datasets mostly contain frequent words that is insufficient for such tests.
- Simlex-999: Simlex-999 dataset (Hill et al., 2015) contains 999 pairs of words whose similarity is annotated by mechanical turkers.

2.12.1.2 Analogy Task

Semantic-Syntactic Word Relationship test set (Google Analogy Task) introduced by Mikolov et al. (2013) consists of pairs of words in the form of a is to a^* as b is to b^* (such as Paris is to France as London is to England). The aim is to find b^*, given a, a^* and b (cosine distance is used as a distance metric to find the missing word). There are 8869 semantic and 10675 syntactic questions in the dataset and the success is measured by accuracy.

2.12.1.3 Synonym Selection Tasks

Given a word, the aim of this task is to select the most synonym-like of the word among the list of candidates. Accuracy (%) is used to measure the performance.

- ESL-50 (Turney, 2001) : Contains 50 synonym selection questions from ESL (English as a second language) tests.
- TOEFL-80 (Landauer and Dutnais, 1997): Contains 80 synonym selection questions from TOEFL (Test of English as Foreign Language) tests.
- RD-300 (Jarmasz, 2003): Contains 300 synonym selection problems from Reader's Digest Power Game.

2.12.1.4 Downstream Tasks

As representations and models get better and the difference between word embedding methods and language models gets closer, experiments are shifted from similarity and relatedness tasks to downstream tasks.

GLUE benchmark dataset (Wang et al., 2018) is introduced to provide a stable testing environment for researchers. It consists of several downstream tasks:

- CoLA: The Corpus of Linguistic Acceptability (Warstadt et al., 2019) is a sentence classification task where the aim is to determine whether a sentence is linguistically acceptable or not. It contains 9594 sentences from linguistic publications and the success is measured by Matthew' Correlation Coefficient (MCC).

- SST-2: The Stanford Sentiment Treebank (Socher et al., 2013) consists of 68.8k sentences from movie reviews. The aim is to classify the sentiment of sentences. Accuracy is used to measure the performance.

- MRPC: Microsoft Research Paraphrase Corpus (Dolan et al., 2004) contains 5800 pairs of sentences from news sources on the web. Each pair is annotated by humans indicating whether they are semantically equivalent or not. Performance is measured by accuracy.

- STS-B: Semantic Textual Similarity Benchmark (Cer et al., 2017) is composed of 8628 pairs of sentences from various sources, annotated between 1 and 5, determining how similar they are. Success is measured by Spearman's rank correlation (ρ).

- QQP: Quora Question Pairs (chen et al., 2018) dataset contains over 400k question pairs where the aim is to determine whether the questions are semantically similar or not. Success is measured by accuracy.

- MNLI: Multi-Genre Natural Language Inference (Williams et al., 2018) dataset is composed of 430k crowd-sourced sentence pairs annotated with entailment information. The aim is to predict whether a second sentence is a contradiction, entailment, or neutral to the first one. Accuracy is used to measure the performance.

- QNLI: Questions Natural Language Inference (Rajpurkar et al., 2018a) dataset is a modified version of the SQuAD dataset (Rajpurkar et al., 2016). It contains over 100k sentence/context pairs where the aim is to determine if the context contains an answer to the question or not.

- RTE: Recognizing Textual Entailment (Bentivogli et al., 2009) is similar to MNLI where the aim is to predict the type entailment of a paragraph and a sentence between entailment, contradiction and unknown.

- WNLI: Winograd Natural Language Inference (Levesque et al., 2012) dataset also concerns with Natural language inference similar to the MNLI and the RTE datasets.

Stanford Question Answering Dataset (SQuAD 1.1 (Rajpurkar et al., 2016) and SQuAD 2.0 (Rajpurkar et al., 2018b)) is a reading comprehension dataset that is composed of wikipedia articles and question related to them. The aim is to find the text segment that gives the answer to the corresponding question. There are 150k question 50k of which is unanswerable from the given context article. Any model built for this task should also determine whether the question is answerable or not in addition to answering the questions.

RACE dataset (Lai et al., 2017) is also a dataset for reading comprehension taken from the English exams for middle and high school chinese students. The aim is to find the correct answer to the questions about a certain text passage, among the choices. There are approximately 28k passages and 100k questions.

Leaderboards of current state-ot-the-art can be tracked either from the respective websites or from ACL Wiki website (`https://aclweb.org/aclwiki/State_of_the_art`). The reader can refer to Bakarov (2018) for comparisons, advantages, and disadvantages of the evaluation methods of word embedding models.

2.12.2 Results

In this section, we report the results obtained by the models examined in this thesis on aforementioned datasets. In Tables 2.6, 2.7, 2.8, and 2.9, results in similarity, analogy, synonym selection, and downstream tasks are given respectively.

While reporting the results, we follow a few criteria to make it as fair and simple as possible:

- Unless noted otherwise, all of the results are taken from the original papers.

- If more than one paper report results on the same model, we take the one in the original paper.

- If the author(s) provide several variations of a model, we report only the one with the best score.

Although some of the differences in performances of word representations are due to the models themselves, it should be noted that the size of the datasets that the models

Table 2.6: Word Embedding Models' Performances in Similarity Tasks (in Chronological Order). Bottom part shows the results of multi-modal embeddings

Model	Dim.	WS-353	SCWS ($\rho \times 100$)					RG-65	MEN	YP-130	RW	MC-30	Simlex-999
		($\rho \times 100$)	avgSim	avgSimC	globalSim	localSim	MaxSimC	($\rho \times 100$)	($\rho \times 100$)	($\rho \times 100$)	($\rho \times 100$)	($\rho \times 100$)	($\rho \times 100$)
HLBL	100	33.2]	-	-	-	-	-	-	-	-	-	-	-
C&W	50	29.5	-	-	57.0	-	-	48.0	57.0	-	-	-	-
C&W	50	49.8	-	-	-	-	-	-	-	-	-	-	-
R&M	-	73.4	60.4	60.5	62.5	-	60.4	-	-	-	-	-	-
RNNLM	640	-	-	-	-	-	-	-	-	-	-	-	-
Huang et al. (2012)	50	71.3	62.8	65.7	58.6	26.1	-	-	-	-	-	-	-
CBOW	400	69.4	64.2	-	-	-	-	73.2	66.5	34.3	-	-	-
Skip-Gram	100	58.9	-	-	-	-	-	-	-	-	-	-	-
Skip-Gram	300	70.4	66.6	66.6	65.2	-	-	-	-	-	-	-	-
Skip-Gram	256	66.7	-	-	-	-	-	-	55.7	-	38.8	-	-
Luong et al. (2013)	50	64.6	-	-	48.5	-	-	65.4	-	-	34.4	71.7	-
CLBL	-	39.0	-	-	-	-	-	41.0	-	-	30.0	-	-
Tian et al. (2014)	50	-	-	65.4	-	-	63.6	-	-	-	-	-	-
Qiu et al. (2014)	200	65.2	-	-	53.4	-	-	67.4	-	-	32.9	81.6	-
MSSG	300	70.9	67.3	69.1	65.5	59.8	-	-	-	-	-	-	-
Chen et al. (2014)	200	-	66.2	68.9	64.2	-	-	-	-	-	-	-	-
GloVe	300	75.9	-	-	59.6	-	-	82.9	-	-	47.8	83.6	41.0
Guo et al. (2014)	50	-	49.3	-	-	-	55.4	-	-	-	-	-	-
KNET	100	66.1	-	-	-	-	-	-	-	-	39.3	-	-
CNN-VMSSG	300	-	65.7	66.4	66.3	**61.1**	-	-	-	-	-	-	-
AutoExtend	300	-	**68.9**	**69.8**	-	-	-	-	-	-	-	-	-
SenseEmbed	400	**77.9**	62.4	-	-	-	-	**89.4**	80.5	**73.4**	-	-	-
TWE-1	400	-	-	68.1	-	-	**67.3**	-	-	-	-	-	-
Jauhar et al. (2015)	80	63.9	-	-	65.7	-	-	73.4	64.6	-	-	75.8	-
SAMS	300	-	62.5	-	59.9	58.5	-	-	-	-	-	-	-
SWE	300	72.8	-	-	-	-	-	-	-	-	-	-	-
Soricut and Och (2015)	500	71.2	-	-	-	-	-	75.1	-	-	41.8	-	-
Cotterell et al. (2016)	100	58.9	-	-	-	-	-	-	-	-	-	-	-
char2vec	256	34.5	-	-	-	-	-	-	32.2	-	28.2	-	-
Bojanowski et al. (2016)	300	71.0	-	-	-	-	-	-	-	-	47.0	-	-
Yin and Schütze (2016)	200	76.0	-	-	-	-	-	-	**82.5**	-	**61.6**	85.7	48.5
dLCE	500	-	-	-	-	-	-	-	-	-	-	-	**59.0**
Ngram2vec	300	-	-	-	-	-	-	-	76.0	-	44.6	-	42.1
MSWE	300	72.4	66.7	66.7	**66.8**	-	-	-	76.4	-	35.6	-	39.2
Dict2vec	300	75.6	-	-	-	-	-	87.5	75.6	64.6	48.2	**86.0**	-
LMM	200	61.5	-	-	63.0	-	-	63.1	-	-	43.1	-	-
LSTMEmbed	400	61.2	-	-	-	-	-	-	-	-	-	-	-
Bruni et al. (2014a)	-	70.0	-	-	-	-	-	-	69.0	-	-	-	-
Collell Talleda et al. (2017)	-	69.4	-	-	-	-	-	-	81.3	-	-	-	41.0

are trained on can be different, therefore can affect the fairness of comparison.

Table 2.6 shows word embedding models' performances in similarity tasks. SenseEmbed (Iacobacci et al., 2015) is the best performing model in WS-353, RG-65, and YP-130 datasets according to the reported results. Yin and Schütze (2016) has superior performance in the datasets of MEN and RW, while Dict2vec (Tissier et al., 2017) outperforms others on MC-30. In SCWS, AutoExtend (Rothe and Schütze, 2015) gives the highest correlation coefficient scores. In general, GloVe (Pennington et al., 2014), SenseEmbed (Iacobacci et al., 2015), Yin and Schütze (2016), and Dict2vec (Tissier et al., 2017) perform well on similarity datasets.

SenseEmbed's (Iacobacci et al., 2015) success can be attributed to its capability to disambiguate senses by being trained on sense-tagged corpora. Glove (Pennington et al., 2014) is generally robust as it's a mixture of global cooccurrence and local context-based methods. When it comes to Yin and Schütze (2016), it is an ensemble of existing embeddings including Glove, which produces better representations for OOV words due to its ensemble nature. Thus, it has a good coverage of words in similarity datasets. Dict2vec's (Tissier et al., 2017) performance proves the effectiveness of positive sampling over word2vec (Mikolov et al., 2013).

Word embedding models' performances are tested on Google Analogy Task that includes both syntactic and semantic analogies (Table 2.7). The best accuracy scores are obtained by Yin and Schütze (2016) in this category. Glove (Pennington et al., 2014)

Table 2.7: Word Embedding Models' Performances in Analogy Task (in Chronological Order).

Model	Dimension	Google Analogy Task (acc. %)		
		Syntactic	Semantic	Total
C&W	50	9.3	12.3	11.0
RNNLM	640	8.6	36.5	24.6
CBOW	1000	57.3	68.9	63.7
Skip-Gram	1000	66.1	65.1	65.6
Skip-Gram	100	36.4	28.0	32.6
Skip-Gram	300	61.0	61.0	61.0
Skip-Gram	256	51.3	33.9	43.6
ivLBL	100	46.1	40.0	43.3
ivLBL	300	63.0	65.2	64.0
vLBL	300	64.8	54.0	60.0
vLBL	600	67.1	60.5	64.1
Qiu et al. (2014)	200	58.4	25.0	43.3
MSSG	300	-	-	64.0
GloVe	300	69.3	81.9	75.0
KNET	100	46.9	24.9	36.3
char2vec	256	52.5	2.5	35.5
Fasttext	300	74.9	77.8	-
Yin and Schütze (2016)	200	**76.3**	**92.5**	**77.0**
Ngram2vec	300	71.0	74.2	72.5
MSWE	50	-	-	69.9
LMM	200	20.4	-	-

Table 2.8: Word Embedding Models' Performances in Synonym Selection Tasks (in Chronological Order).

Model	Dimension	ESL-50 (%)	TOEFL-80 (%)	RD-300 (%)
Skip-Gram	300	-	83.7	-
Skip-Gram	400	62.0	87.0	-
GloVe	300	60.0	88.7	-
MSSG	300	57.1	78.3	-
Jauhar et al. (2015)	80	63.6	73.3	66.7
Jauhar et al. (2015)	80	**73.3**	80.0	-
Li and Jurafsky (2015)	300	50.0	82.6	-
SWE	300	-	88.7	-
LSTMEmbed	400	72.0	**92.5**	-

follows it as the second best performing model. Results in Google Analogy task can be interpreted much as those in similarity tasks.

In synonym selection tasks, Jauhar et al. (2015) provides the best results in ESL-50 dataset while LSTMEmbed (Iacobacci and Navigli, 2019) performs the best in TOEFL-80 dataset. Results can be seen in Table 2.8

In Table 2.9, word embedding models' performances on downstream tasks are provided. In GLUE benchmark, CBOW (Mikolov et al., 2013), BiLSTM+Cove+Attn (McCann et al., 2017), and BiLSTM+Elmo+Attn (Peters et al., 2018) are behind human baselines except for the task of QQP. In QQP, CBOW is still underperforming but BiLSTM+Cove+Attn (McCann et al., 2017) and BiLSTM+Elmo+Attn (Peters et al., 2018) are superior to human performance.

Table 2.9: Word Embedding Models' Performances in Downstream Tasks.

Model	CoLA (mcc)	SST-2 (%)	MRPC (F1)	STS-B ($\rho \times 100$)	QQP (F1)	MNLI m/mm (%/%)	QNLI (%)	RTE (%)	WNLI (%)	SQuAD 2.0 (F1)	RACE (%)
CBOW [1]	0.0	80.0	81.5	58.7	51.4	56.0/56.4	72.1	54.1	62.3		
BiLSTM+Cove+Attn [1]	8.3	80.7	80.0	68.4	60.5	68.1/68.6	72.9	56.0	18.3	-	-
BiLSTM+Elmo+Attn [1]	33.6	90.4	84.4	72.3	63.1	74.1/74.5	79.8	58.9	65.1	-	-
GLUE Human Baselines	66.4	97.8	86.3	92.6	59.5	92.0/92.8	91.2	93.6	95.9	-	-
SQuAD Human Baselines	-	-	-	-	-	-	-	-	-	89.4	-
Turkers (Lai et al., 2017)	-	-	-	-	-	-	-	-	-	-	73.3
BERT	60.5	94.9	89.3	86.5	72.1	86.7/85.9	91.1	70.1	65.1	89.1[12]	72.0[12]
ERNIE 2.0	63.5	95.6	90.2	90.6	73.8	88.7/88.8	94.6	80.2	67.8	-	-
XLNet (ensemble)	67.8	96.8	92.9	91.6	**74.7**	90.2/89.7	98.6	86.3	90.4	89.1	81.8
RoBERTa (ensemble)	67.8	96.7	92.3	91.9	74.3	90.8/90.2	98.9	88.2	89.0	89.8	83.2
ALBERT	**71.4**	96.9	90.9	**93.0**	-	90.8	95.3	89.2	-	90.9	86.5
ALBERT (ensemble)	69.1	**97.1**	**93.4**	92.5	74.2	**91.3/91.0**	**99.2**	**89.2**	**91.8**	**92.2**	**89.4**
GPT-3 Few-Shot -	-	-	-	-	-	-	-	69.0	-	69.8	45

As for the original BERT (Devlin et al., 2019) and its variants, in the tasks of MRPC, QQP, QNLI they consistently outperform human baselines. In SST-2, MNLI, RTE, and WNLI, human performance is better. In STS-B, the only model with superior performance to humans is ALBERT (Lan et al., 2020), In CoLA, and the tasks of question answering (SQuAD 2.0) , and reading comprehension (RACE), starting from XLNET (Yang et al., 2019) better performances over humans are observed. GPT-3 (Brown et al., 2020) is promising with its language model meta-learner idea and gives its best performance in the Few-Shot setting. Although it is behind the state-of-the-art by a large margin in GLUE benchmark, in RTE its score is beyond CBOW (Mikolov et al., 2013), BiLSTM+Cove+Attn (McCann et al., 2017), and BiLSTM+Elmo+Attn (Peters et al., 2018). Table 2.9 proves the success of contextual representations especially the transformer-based models (BERT (Devlin et al., 2019) and its successors) by going beyond human performance in most of the downstream tasks. However, it can be said that in natural language inference tasks such as MNLI, WNLI, and RTE, these probabilistic language representations still have some limitations in meeting causal inference requirements.

Table 2.9 proves the success of contextual representations especially the transformer-based models (BERT (Devlin et al., 2019) and its successors) by going beyond human performance in most of the downstream tasks. However, it can be said that in natural language inference tasks such as MNLI, WNLI, and RTE, these probabilistic language representations still have some limitations in meeting causal inference requirements.

[1]reported as baselines in GLUE (Wang et al., 2018)

CHAPTER 3

METHOD

In this chapter, the details of the proposed model and dataset are introduced. First, a newly created dataset from Wikimedia Commons is explained in Section 3.1, in detail. In the following Sections 3.2 and 3.3, individual parts of the model are explained respectively, for both text processing and image processing. Methods for combining those text and image parts are explained in Section 3.4. Finally, the last section introduces the training method of the combined model. While Section 3.5.1 introduces the pre-training of the multi-modal language model, Section 3.5.2 describes the fine-tuning steps where the model is trained and tested on downstream tasks.

3.1 Wikimedia Commons Dataset

Wikimedia Commons[1] is a repository of free-to-use images that is a part of Wikimedia Foundation. Files from Wikimedia Commons are used across all Wikimedia projects in all languages, including Wikipedia, Wiktionary, Wikibooks, Wikivoyage, Wikispecies, Wikisource, and Wikinews, or downloaded for offsite use. It is comprised of approximately 65 million images that take approximately 250 TB space. In addition to the images, they also contain useful information such as caption, description, and the timestamp of the images.

In order to retrieve the images, queries must be sent to Wikimedia Commons website. To this end, we have used two different sets of query words to construct datasets. For retrieving the entire dataset, the dictionary of the BERT model (Devlin et al., 2019) is used. For retrieving the subset that we mostly used in this work, the words in UWA MRC psycholinguistic dataset are used (explained in detail in Section 4.1.1). Each word in that dataset contains a score of concreteness between 100 and 700 where 700 means very concrete and 100 means very abstract. Therefore, we end up with images and their corresponding captions and descriptions labelled with a level of concreteness.

For each word mentioned above, a query is sent to the Wikimedia Commons

[1]https://commons.wikimedia.org/wiki/Main_Page

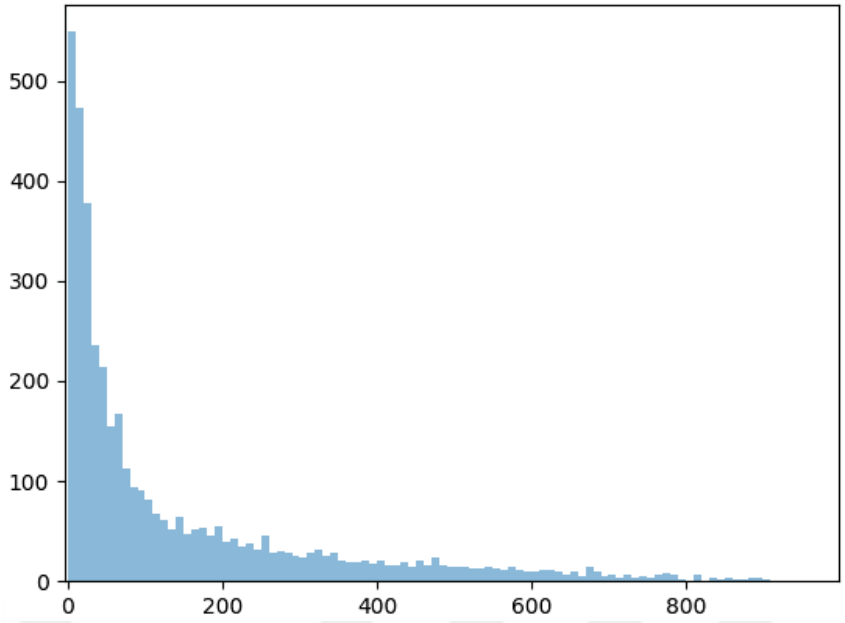


Figure 3.1: Histogram of samples retrieved for words. Horizontal axis shows the number of images retrieved while the vertical axis shows the amount of words which have that many images associated with them.

website with 1000 as a maximum threshold of results for each word. Figure **??** shows the number of words in UWA MRC psycholinguistic dataset and their corresponding sample sizes. As can be seen from the graph, most of the query words returned less than 100 results despite a large threshold. Only around a hundred words have more than 500 images associated with them. The number of samples collected can be seen in Table 3.1. More than 43 Million images are collected using the dictionary of BERT, while approximately 3.2 images are collected using the words in UWA MRC psycholinguistic dataset. It can also be observed that not all images have a description and/or caption associated with them. Some images contain only captions, some images contain descriptions but no caption and finally, some images do not contain any textual information at all. In total, 630k images contain captions and approximately 2M images contain descriptions. As has been described above, there is an overlap between both sets which means that some images contain both caption and description.

Retrieved images have many formats such as .jpeg, .jpg, .jpe .png, .apng, .gif, .tif, .tiff, .xcf, .webp and many image modes such as RGB (3x8-bit pixels, true color), CMYK (4x8-bit pixels, color separation), I (32-bit signed integer pixels), I;16 (16-bit unsigned integer pixels). Although many of these formats and modes are supported, some of them

Table 3.1: Wikimedia Commons dataset statistics

Dataset	# of images	# of captions	# of descriptions
Complete Dataset	43,726,268	1,022,829	17,767,000
Subset (queried w/ UWA MRC words)	3,206,765	629,561	1,961,567

Table 3.2: Wikimedia Commons dataset statistics after filtering.

Dataset	# of images	# of captions	# of descriptions
before preprocessing	3,206,765	629,561	1,961,567
after preprocessing	-	603,089	-
abstract	-	177,308	-
concrete	-	425,781	-
Average word count	-	8.80	42.79

needed to be eliminated. Images with the extension .xcf and .webp are filtered because they are not supported by any of the mainstream image processing libraries. In addition to this, images with mode I (and other modes of I such as I;16, I;16L, I16B and so on) are eliminated because they are single-channel image modes and the neural network models that process these images run with multi-channel inputs. Nearly 26k images are eliminated after this filtering. Dataset statistics after applying the filters can be seen in Table 3.2. In the final version of the dataset, there are approximately 600k images with captions where 177k belongs to abstract concepts while 425k belongs to concrete concepts.

Many images in Wikimedia Commons have a very high resolution (resolutions such as $3000x5000$, $6000x6000$ are very common) therefore require huge storage space. In addition to the filters applied above, a resize operation is performed on images after the download is completed to cope with this storage problem. All images are converted to a resolution of 224x224 since all the image models (AlexNet (Krizhevsky et al., 2012), GoogleNet (Szegedy et al., 2015), VGG (Simonyan and Zisserman, 2015), Resnet (He et al., 2016)) run with those.

Figure 3.2 shows some example images and their corresponding captions and descriptions from the collected Wikimedia Commons dataset. Examples are selected from images that contain both a caption and a description, except for the bottom-left image where a description does not exist.

One thing to be observed from these images is, indeed the images and the texts convey different information on the relationship of concepts. For example, in the top-left image, there is no textual information, neither in the caption nor in the description, about the buildings that can be seen in the image. But streets are mostly located near buildings[2], which is captured by the image. Therefore the system can learn a relationship of concrete concepts such as "street" and "building" from the images without relying on the text. Similarly, the image contains no definite information about where it is taken, but it is understandable from both the caption and the description that it is in Mogadishu, Somalia. The same thing can also be seen in the bottom-left image; there is no mention of a sea/lake in the text but the lighthouse and the sea/lake can be seen together (which occur with almost no exception in real life) in the image which will help the model to learn their

[2]almost 70% of all images from Wikimedia Commons contains buildings when you search for the keyword "street"

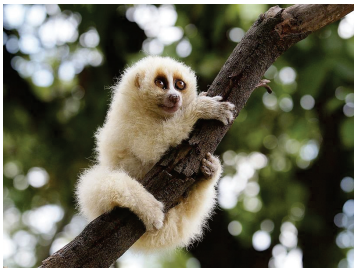
		
Caption	A man carries a huge hammerhead through the streets of Mogadishu	Sheep lounging in the shade of a tree with matriarch standing guard
Description	Mogadishu, Somalia. 10/10/2015. A man carries a huge hammerhead shark through the streets of Mogadishu. A recent escalation of plunders of Somali waters by foreign fishing vessels could mean the return of hijackings, locals warn. The country's waters have been exploited by illegal fisheries and the economic infrastructure that once provided jobs has been ravaged. Somalia has been at war for the last 25 years, but 2017 is a turning point. This country in the Horn of Africa is holding its first free elections since 1969; a whole culture is being overturned. Those who created it have shot and killed, but finally, they are on the losing side.	A flock of sheep (*Ovis aries*) lounging in the shade of a tree with the matriarch of the flock standing outside the shade. The flock was kept in the enclosed area of Röe Castle ruin to keep the vegetation in check. The standing matriarch is tagged in both ears meaning that she is selected for breeding and will not be slaughtered after her first year. The rest of the flock have tags in only one ear and will be slaughtered withing twelve months after their birth. Röe Castle ruin, Röe, Lysekil Municipality, Sweden. The image is stacked manually from two photos (handheld) for focus and light.
		
Caption	Aniva lighthouse on a rocky promontory in Sakhalin, Russia, with a flock of gulls circling in the surrounding mists	A Javan Slow Loris (Nycticebus javanicus) clings to a branch.
Description	-	The Javan slow loris (*Nycticebus javanicus*) is a strepsirrhine primate and a species of slow loris native to the western and central portions of the island of Java, in Indonesia. Although originally described as a separate species, it was considered a subspecies of the Sunda slow loris (*N. coucang*) for many years, until reassessments of its morphology and genetics in the 2000s resulted in its promotion to full species status. It is most closely related to the Sunda slow loris and the Bengal slow loris (*N. bengalensis*). The species has two forms, based on hair length and, to a lesser extent, coloration.

Figure 3.2: Example images and their corresponding captions and descriptions from the Wikimedia Commons Dataset.

relationships better. So, a language model trained with both images and text can help to improve the performances of language models.

Although both captions and descriptions are collected within the dataset, mostly captions are used to train the multi-modal language models because of two main reasons. Descriptions in Wikimedia Commons is observed to be unclean. There are many additional texts which need to be cleaned extensively, such as copyright notices, information about photographer or information about how the photograph is taken (such an example can be seen in the last sentence of the top-right image of Figure 3.2). On the other hand, captions are already cleaned and only contain information about the picture itself. Because of the requirement of tedious cleaning, captions are easier to use.

The second but the most important reason is the image-text alignment issues. Captions are written in a way that describe the images briefly without giving any other information or making any other comment that can be classified as common-sense knowledge or real-world knowledge. Contrarily, descriptions contain much information that cannot be seen in or referred from the images. Although these additional pieces of knowledge can be important and useful in other tasks, in language modelling, they break the image-text alignment and lead to learning noisy contexts. If we take the top-right image in Figure 3.2 as an example, we can see how this can affect the language models. Description of the top-right image provides many semantically similar words such as "breeding", "slaughtered" and "vegetation" to the context of the image which is sheep lounging in a field. But it also provides a lot of dissimilar or unrelated words such as "castle", "ruin", "municipality" which has very little to do with the image itself. Consequently, this leads to learning from an accidental relationship, for example, between the context of "sheep" and the context of "municipality". On account of this fact, captions are used in all language modelling tasks in this work to provide a better image-text alignment in training samples.

There have been several other multi-modal datasets proposed in the literature that consist of image-text pairs such as Flickr (Young et al., 2014), MS COCO (Sharma et al., 2018), Wikipedia, British Library, and ESP Game (von Ahn and Dabbish, 2004). Table 3.3 shows the collected dataset in comparison with these multi-modal datasets. The Flickr dataset and MS COCO dataset contain image-caption pairs, while the Wikipedia dataset provides the images in Wikipedia with their corresponding articles. The British Library book dataset, on the other hand, contains historical books and the pictures depicted in them. Finally, the ESP game dataset consists of 5 words for each image labeled by human annotators. Although both Wikipedia and BL datasets provide much longer texts, they lack the image-text alignment of caption datasets. Therefore, caption datasets such as MS COCO, Flickr, or the proposed dataset in this work are more suited to the task of multi-modal language modeling. Compared with these image captioning datasets, the size of the collected dataset is much greater. As deep neural representations have massive data requirements, it is preferable to have such a large amount of data. Recently, the WIT dataset

Table 3.3: Comparison of Wikimedia Commons to other multi-modal datasets.

Dataset	# of Images	Textual Source	Ave. Word Length	Additional Info.
Flickr (Young et al., 2014)	32K	Captions	9	-
COCO (Sharma et al., 2018)	123K	Captions	10.5	-
Wikipedia	549K	Articles	1397.8	-
BL	405K	Books	2269.6	-
ESP(von Ahn and Dabbish, 2004)	100K	Object Annotations	5	-
WIT(Srinivasan et al., 2021)	11.4M	Captions/Articles	-	-
	3.98M	Captions/Article (En)	-	-
	568K	Captions (En)	-	-
Wikimedia Commons (ours)	3.2M	-	-	Concreteness Ratings
	629K	Captions	10.2	Concreteness Ratings
	1.96M	Descriptions	57.4	Concreteness Ratings

(Srinivasan et al., 2021) is also proposed with a large number of image-text pairs that can be used for multi-lingual, multi-modal pre-training. It contains 11.4M unique images with captions and descriptive text from Wikipedia articles for various languages. Among them, 3.98M images have textual information in English, where 568K of them have captions. In addition to captions, the collection also includes contextual data such as page titles, page descriptions, section titles with their descriptions. But, the most significant benefit of the proposed dataset is the concreteness labels provided for each image-text pair which might be very useful for various tasks, especially for the multi-modal language modeling. The other datasets mentioned in this section, including WIT, do not contain that information.

3.2 Text Processing Model

In this work, BERT is mainly used for processing text input while DistilBERT is also utilized in some of the tests. In this section, these two models will be presented along with the reasons for their selection and text pre-processing methods.

BERT (Devlin et al. (2019)) is a neural network model that uses a bidirectional transformer architecture (Vaswani et al. (2017)), a self-attention mechanism, to learn contextual word embeddings. Figure 3.3 briefly shows the architecture of BERT. It has multiple layers of transformers (12 in BERT-base, 24 in BERT-large) where each of these individual layers has a hidden layer of size 768 and 1024 respectively for BERT-base and BERT-large, and separate attention heads (12 in BERT-base, 16 in BERT-large). Each attention head spans the entire sentence from both right-to-left and left-to-right, learning "where to look" by producing probabilistic weights for each word.

Different from the earlier language modelling approaches, BERT does not use next word prediction as an objective. Instead, it uses two training objectives: Masked Language Modeling (MLM) and Next Sentence Prediction (NSP). For the MLM objective, randomly selected words are occluded from the model and labelled as masks. Attention heads do not span these masked words since it would create a bias for the prediction. Using

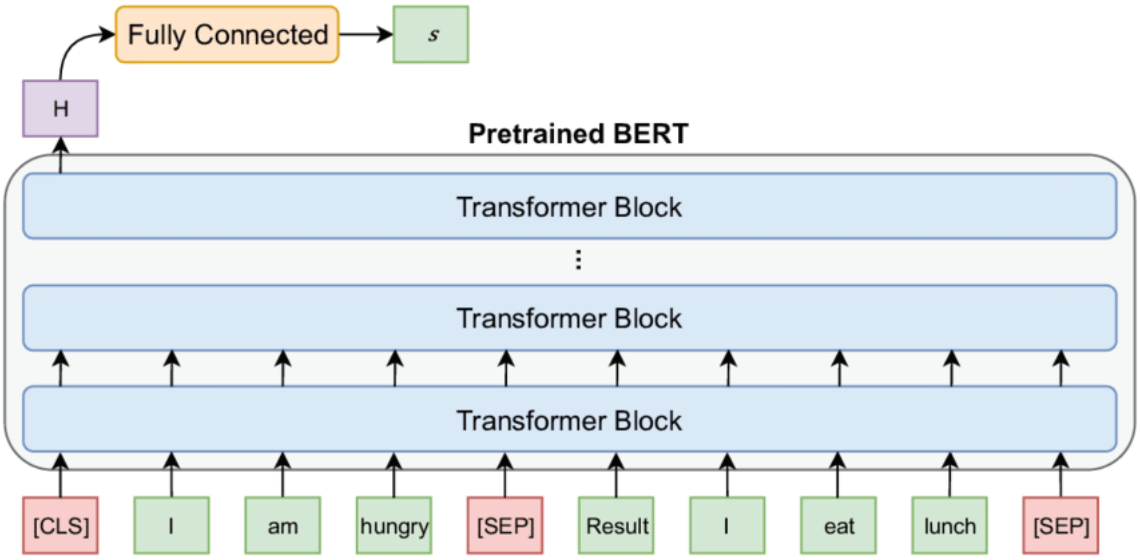


Figure 3.3: BERT model architecture (Devlin et al., 2019). Taken from He et al. (2020).

MLM enables the model to learn contextual dependencies among words very successfully, since the word embeddings of a word are computed depending on the surrounding words, instead of using the same vector in embedding space for every instance of that word. For the NSP objective, the model tries to predict whether the two sentences provided to the model belongs to the same context or not. It helps BERT to consider multiple sentences as context and to represent inter-sentence relations.

In addition to the token (word) embeddings, BERT also uses segment (sentence) embeddings and position embeddings (words' position in segments) as input. The input structure of the BERT can be seen in Figure 3.4. While the sentence embedding determines which sentence the word is in, positional embeddings provide information to the system about the word order. A word's embedding is therefore fed to the model as the average of its token embedding, sentence embedding, and positional embedding. This input structure has many benefits: positional embeddings raise the awareness of the model to word order while segment embeddings help the model with the NSP objective. Also, giving multiple

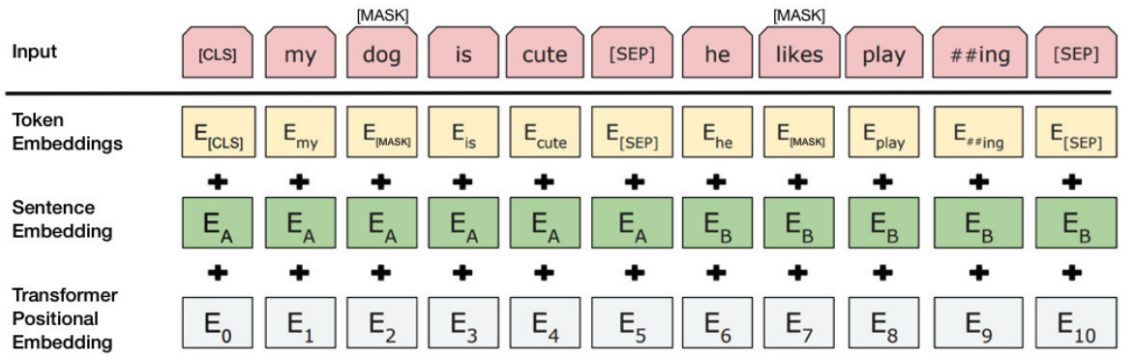


Figure 3.4: BERT model input structure (Devlin et al., 2019). Taken from the original article.

sentences as input helps BERT to be integrated into most downstream tasks that require inter-sentence connections such as Question Answering and Natural Language Inference (NLI) easily without requiring any other architecture.

To integrate BERT to downstream tasks, an additional fully connected layer is used on top of transformer layers to predict the given text's class instead of predicting the target (masked) word (can be observed in Figure 3.3). Usually, the Wikipedia dataset is used to pre-train the model on MLM and NSP objectives, then the resulting parameters are fine-tuned on the downstream task with the addition of the aforementioned fully connected layer.

Some tests that are performed in this study also involve the DistilBERT language model. DistilBERT (Sanh et al. (2019)) is based on the original BERT model. It is a more efficient version of BERT in expense for a small deficiency in classification performance. It retains 97% of BERT's performance while using 40% fewer parameters. To accomplish this, they use knowledge distillation, where a small model is trained to reproduce the behaviour of a larger model (DistilBERT and BERT, respectively, in this case). Knowledge distillation aims to make the student model (DistilBERT) predict the same values as the teacher model (BERT) using fewer parameters. This way, the knowledge learned by the teacher model can be transferred to more efficient student models. Parameter reduction from BERT to DistilBERT comes from the removal of some of the transformer layers in BERT. Authors of DistilBERT show that some of the parameters of BERT are not used in the prediction, therefore, do not contribute to learning downstream tasks. Consequently, they suggest removing some layers and use the knowledge distillation technique to create a more efficient language model.

3.3 Image Processing Model

Resnet (He et al., 2016) is used in this work as an image model mostly due to its success in many image processing tasks. It is a very deep neural network model that relies on Convolutional neural network architecture. At the time it is published, it was the state-of-the-art model in ImageNet (Deng et al., 2009) object classification challenge.

Resnet has several different variations in terms of network depth: 34-layered model Resnet34, 50-layered model Resnet50, 101-layered model Resnet101, and finally the largest model with 152-layers Resnet152. Each layer consists of several 1x1 and 3x3 convolutions. Each model starts and end with an average pooling operation before the first layer and after the last layer. The architecture of Resnet34 can be seen in Figure 3.6.

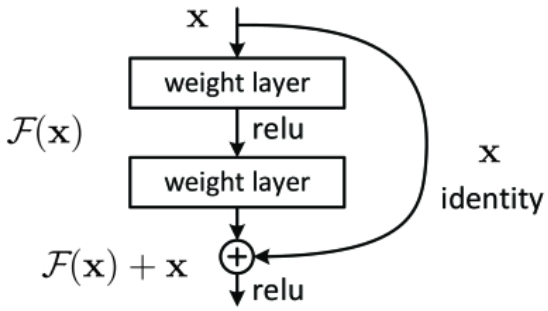


Figure 3.5: Residual Connection. Taken from the original article.

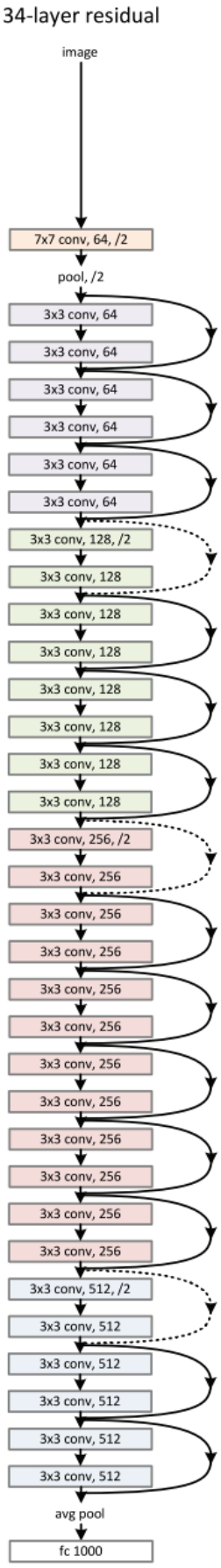


Figure 3.6: Resnet34 Architecture. Taken from the original article.

Stacking so many layers in deep neural networks naively does not immediately lead to better results, instead, it causes performance degradation problems. When the depth of a model increases, training errors increase and accuracy is saturated. To work around this issue and build substantially deeper networks, authors needed a workaround. Therefore, shortcut connections called *residual connections* (see in figure 3.5), are used. These shortcut connections are used after every two layers in the architecture, propagating the inputs to the outputs of those two layers. They are *parameter-free*, which means that they do not perform any operation on the inputs such as pooling, convolution or multiplication, therefore they do not contain any learnable parameters. They merely propagate the inputs to the outputs where they are averaged. Shortcut connections have been proposed by many scientific work with little variations. Its theoretical foundations are first discussed in Ripley and Hjort (1995) and Bishop (1995). On the practical level, highway networks (Srivastava et al. (2015a) and Srivastava et al. (2015b)) and the inception model of GoogleNET (Szegedy et al., 2015) were the first works that leveraged shortcut connections. It is shown that these shortcut connections can overcome the performance degradation problem in very deep neural network architectures making models such as Resnet very successful at stacking many layers and capturing more features than prior models.

In this work, Resnet152 is used because it is shown to outperform the smaller Resnet models and the Wikimedia commons dataset was large enough to tune such a large model.

3.4 Text-Image Combination Methods

Combining multiple modalities can be very problematic and carry the risk of breaking the learned semantic relationship of words by individual models if it does not work the way they are intended to be. To this extend, many studies in this field focuses on the fusion of modalities rather than the individual models. In order to combine the text and vision parts of the model, multiple methods are used in this study. The main method proposed in this work is attentive pooling networks (Santos et al., 2016). Simple averaging of both classifiers and combining them with fully connected layer(s) are also experimented upon to show the effectiveness of the attentive pooling mechanism.

Simple Averaging: This is the simplest form of model combination. In the case of simple averaging, each model is run separately and the predictions are averaged to form the final multi-modal predictions. No additional parameters are used and models are trained separately. Thus, the features learned by models cannot affect each other in any way.

Fully connected Layer(s): The second method of combination is realized through

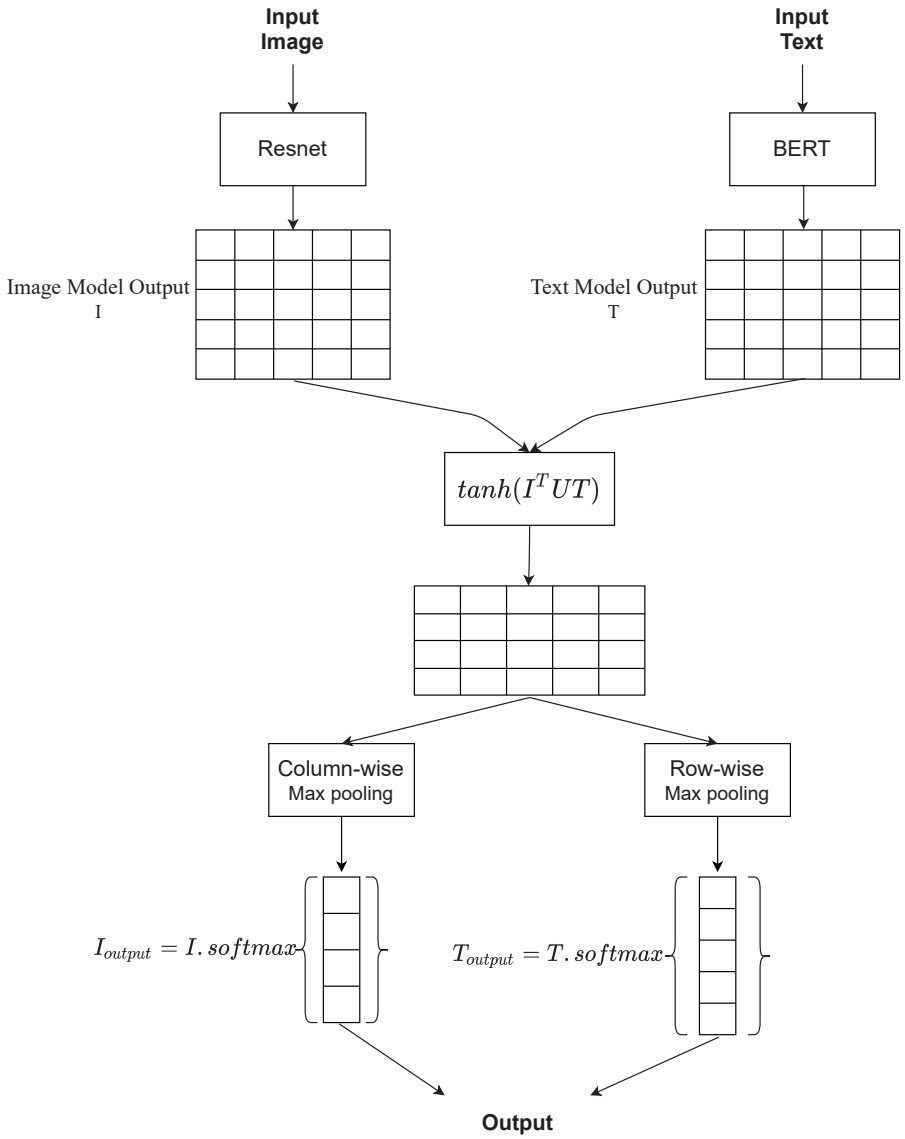


Figure 3.7: Attentive Pooling Networks (Santos et al., 2016).

additional fully connected layer(s) on top of the individual models (text and vision). In this method, outputs from the last fully connected layer in Resnet and the hidden layer outputs of the last transformer layer in BERT are concatenated. Afterwards, the resultant vector is reduced to the number of classes through fully connected layer(s). The number of fully connected layers and their sizes are hyperparameters to the model.

Attentive Pooling Networks: Figure 3.7 shows the model architecture with attentive pooling as a combination mechanism. It is a two-way attention mechanism that is aware of both modalities and jointly learns to attend over them through matrix multiplications and pooling operations.

Attentive pooling takes the hidden states of each word in BERT as textual input and takes the last layer of Resnet in the form of a matrix as visual input. These inputs are multiplied with the matrix U which is composed of parameters to learn and passed through $tanh$ activation. The result is a single matrix that is composed of visual features on the

rows and textual features on the columns. This representation scheme allows features from different modalities to be jointly represented in a single matrix where max-pooling operation is performed over each row and column to find out the most important feature that is also dependent upon the other modality. Two vectors, I_{output} and T_{output}, are the outputs of the attentive pooling mechanism. For fine-tuning this model on downstream tasks, these two outputs are concatenated and passed through an additional fully connected layer to reduce the dimension to the number of classes.

3.5 Multi-Modal Language Model Training

Starting from the contextual language models and onward, the trend for building language models is to pre-train them for language comprehension objectives first, then fine-tuning them by training them on down-stream tasks.

3.5.1 Multi-Modal Language Model Pre-training

The idea of pre-training the neural language models is borrowed from the advances in image processing models (Howard and Ruder, 2018). It is shown in both vision and text models that pre-training the model on a preliminary image understanding/text understanding tasks improves the performance vastly.

For image processing, the pre-training task is usually the object classification task on the ImageNET dataset (Deng et al., 2009). ImageNET dataset has 1.2 million images that are hand-labelled into 1000 categories. Respective models are trained to predict the objects in each image by adding a fully connected layer on top to reduce the feature vectors' size to 1000. The aim here is to teach the model basic image understanding: being able to identify objects and entities in images. It is shown by many vision models that they are even able to differentiate images of 120 different dog breeds in the imageNET dataset such as "Australian terrier" and "Airedale terrier". They manage to do this by understanding what is in the picture by using the shapes and colours of entities in the pictures. The AlexNET model (Krizhevsky et al., 2012) showed that the vision models can learn various colour and shape patterns by probing the learned parameters in their model.

The process is similar for language models with the only difference of pre-training objectives. Earlier models (before BERT) used next word prediction in very large unlabeled text such as Wikipedia and common crawl text. The aim was to predict the next

word given the previous words. Starting from BERT and onward, the pre-training objective changed from the next word prediction to masked language modelling. Using very large datasets that contain billions of words, this method allowed the text models to grasp language understanding successfully. They were able to learn the meaning and semantic/syntactic relations of words (due to distributional hypothesis) which are fundamental to any downstream task.

In this work, the pre-training is realized through the objectives that are inspired by the advances in cognitive psychology. It is shown that language acquisition in children starts with experiential information and continue with textual information (Andrews et al. (2009) and Vigliocco et al. (2009)). As Kiela et al. (2015) state, perceptual information is more relevant for, e.g. elephant than it is for happiness. In other words, we first learn the language through images and learn concrete concepts, then we start learning abstract concepts from textual sources.

Advancements in computational linguistics also reinforce this idea by showing that concrete examples in language are easier to learn while abstract ones are more challenging. Hessel et al. (2018) show that the more concrete the downstream task gets, the easier it becomes for language models. Bruni et al. (2014a) show that the semantic/syntactic similarities of concrete examples on the MEN dataset are easier to learn while the abstract words can get ambiguous. They prove this by showing that the concrete examples have a 0.78 Spearman correlation rank while the abstract examples have 0.52 (Contributing to an overall of 0.76).

To adopt this learning scheme to this project, the Wikimedia Commons Dataset (see Section 3.1) is divided into two categories: Abstract samples and concrete samples. Concreteness levels of words from the UWA MRC Psycholinguistic Database are used to accomplish that (MRC dataset is explained in Section 4.1.1 and Section 3.1 explains how the samples are divided). First, the image model is trained with concrete samples, then the textual model is trained with all of the samples concrete and abstract combined, in a curriculum learning fashion (Bengio et al., 2009). Therefore, the learning model in humans is mimicked through this pre-training process.

3.5.2 Multi-Modal Language Model Fine-tuning

Once the pre-training objective is completed and the image/text model gained basic image/language understanding respectively, the last fully connected layer is removed from the model and replaced with an appropriate classification layer according to the task at hand. The model is, then, fine-tuned for the downstream task. For image models,

downstream tasks can be object detection, semantic segmentation, etc. while on the textual models they are composed of sentiment analysis, sentence classification, natural language inference, and so on. There are also several tasks that can bring visual and textual information together such as Visual question answering and image captioning. Such tasks require multi-modal solutions where a pre-trained image model is combined with a pre-trained text model.

In this work, a similar approach is taken on multi-modal tasks where each model is pre-trained on the task at hand, then combined through various methods (see Section 3.4) to produce feature vectors. Those feature vectors are, in turn, reduced to the number of classes through the use of fully connected layers. But, unlike most models (VilBERT (Lu et al., 2019) being the only exception), additional multi-modal pre-training step is applied using abstract and concrete samples in a curriculum learning fashion. To accomplish that, first, the image model is pre-trained with concrete samples, then the text model is pre-trained with the abstract samples, mimicking the language acquisition in humans. The next chapter introduces the tasks used in this work in detail while discussing the results obtained with the model and curriculum learning scheme introduced in this chapter.

CHAPTER 4

EXPERIMENTS

4.1 Datasets

Before delving deep into details about the experimental results, the datasets used in this work are explained here in detail. Section 4.1.1 describes the UWA MRC Psycholinguistic Database proposed by Coltheart (1981) and Section 4.1.2 introduces Visual Question Answering (VQA) dataset of Antol et al. (2015).

4.1.1 UWA MRC Psycholinguistic Database

MRC Psycholinguistic Dataset (Coltheart, 1981) is a dataset, produced and maintained by researchers in University of Western Australia (UWA). The dataset contains 98538 words and their properties such as: *type*, *meaningfulness*, *concreteness*, *part-of-speech*, *familiarity*, and many more. It is a combination of many different smaller datasets produced by many different research. Concreteness scores which are used in this research are derived from merging the two datasets provided by Paivio et al. (1968) and Gilhooly and Logie (1980).

In this dataset, 4293 out of 98538 words have a concreteness rating, rated by human annotators. Human annotators are asked to rate the concreteness of words between (including) 1 and 7 where higher the score, more concrete the word is, and vice versa, the lower the score, more abstract the word is. The mean of all users is used as the final concreteness rating of the word between 100 and 700 (least significant two digits come from the floating points in mean value. i.e. if the mean of a word's concreteness is 3.452, its concreteness score is 345). Overall, the most abstract word in the dataset is "as" with a rating of 158, and the most concrete word is "milk" with a rating of 670. The mean rating of all words is 438 and the standard deviation is 120.

The dataset is divided into two parts: abstract words and concrete words using 400 as a threshold. All the words below 400 rating is considered as abstract and all the words

Table 4.1: Example words and their concreteness scores in UWA MRC Psycholinguistic dataset.

Word	Concreteness Rating	Classification
As	158	
Apt	183	
Impossible	198	
Humble	231	
Maturity	234	Abstract
Optimism	240	
Research	366	
Math	386	
Midnight	396	
She	406	
Weather	439	
Shiver	455	
Undergraduate	500	
Equipment	532	Concrete
Shield	576	
Ox	633	
Tomato	662	
Milk	670	

above that value is considered as concrete. Table 4.1 shows some example words, their ratings, and labels. It can be seen that the most concrete words are usually nouns that have distinct properties with little to no variation such as "milk" (all milks are white and liquid), "tomato" (almost all tomatoes are red or very close to red in color and have very similar shapes), and "ox". On the opposite side of the spectrum, we mostly see adjectives such as maturity, humble, and impossible that are mostly used as properties. They are very hard to define and can vary a lot in meaning depending on what they refer to.

When we look at the middle of the spectrum of concrete words, we see words such as "equipment", "shield", and "undergraduate". Although all of these words refer to concrete objects which are easily observable and identifiable, they can be in very different forms and shapes, making them more vague than the words such as "tomato" and "milk". Such examples are given in Table 4.2 and 4.3. While the hypernym word "equipment" has a concreteness of 532, its hyponyms are rated much higher with "mallet" being 623 and "hammer" being 605. Similarly, we can observe this relationship with action/performer pairs too: while the word "speak" has a lower concreteness rating, "speaker" and "speech" have much higher ratings due to the fact that they are outputs of the action "speak" and remove some ambiguity.

Words like "weather", "she", "midnight", and "math" constitute the close-to-neutral part of the spectrum. It can be argued that although we can observe a lot of things that either direct consequences or part of those words, it is very hard to define the boundaries

Table 4.2: Concreteness scores of words with hypernym/hyponym relation.

Word	Concreteness Rating
Equipment	532
Hammer	605
Mallet	623

Table 4.3: Concreteness scores of words with action/performer relation.

Word	Concreteness Rating
Speak	419
Speech	453
Speaker	537

Table 4.4: UWA MRC Psycholinguistic Database statistics.

	Abstract	Concrete	Total
Before stop-word removal	1851	2442	4293
After stop-word removal	1674	2434	4108

of the definition of those words, making them neutral in nature.

Qualitative examples above show that, the dataset is successful at capturing the concreteness levels of words in language. But, in order to successfully integrate this dataset into our task some processing is required. Although the UWA MRC Psycholinguistic dataset is successful at identifying the concreteness of words, they consider the words in isolation unlike this work, where contextual embeddings and language models are used to consider words in their context. Therefore, all the stop-words are removed[1] from the dataset considering that they can appear in various contexts with different levels of concreteness and therefore can lead to misleading results. It is observed from the dataset that the lowest rated words are usually stop-words such as "as", "therefore", "and", and so on. This leads to the removal of a lot of abstract words in the lower bound. The most abstract word in the dataset after the removal is "apt" with a rating of 183. Table 4.4 shows the word counts of the dataset before and after the removal of stop-words. The final version of the dataset contains 1674 abstract words and 2434 concrete words.

4.1.2 Visual Question Answering Dataset

Visual Question Answering dataset is a multi-modal dataset that is proposed by Antol et al. (2015). It includes approximately 200k images from COCO dataset (Sharma et al., 2018) paired with approximately 600k questions (3 for each image). Each image in this dataset has multiple questions associated with it in various forms such as yes/no questions and open-ended questions (see Figure 4.1). Yes/No questions are binary questions such as "Is the umbrella upside down?", while the open-ended questions such as "Who is wearing glasses?" require more diverse answers. Close to 40% of all questions are

[1]Stop-words from NLTK library are used.

Figure 4.1: Some questions in VQA dataset (Antol et al., 2015).

yes/no questions, where approximately 58% of answers is yes, and the rest is open-ended. Open-ended questions have a variety of types including but not limited to "What is...?", "Is there...?", "How many...?" and "Does the...?".

As it can be seen from Figure 4.1, questions require a lot of inference on objects in terms of both intra and inter modality. For example, the top question in the aforementioned figure requires the identification of man, women, and glasses, both visually and semantically, to match their representations with the other modality and infer the answer by calculating the word "wearing" through proximity of objects in images. Similarly, the bottom image requires the identification of the word umbrella in the image along with its orientation.

Although the dataset requires a lot of inference between modalities, Agrawal et al. (2018) state that the dataset includes bias towards some question/answer pairs. In their work, they show that questions related to colours ("What is the colour of ...?" or "is ... white?") almost always lead to the answers of white/no for open-ended and yes/no question respectively. Similarly Goyal et al. (2017) suggest that answering the questions that are starting with the phrase "Do you see a ...?" with yes blindly, leads to an accuracy of 87% among those questions. Therefore, using language priors alone, a model can correctly predict a significant amount of questions. Human baselines (Table 4.5. Calculated from $3k$ samples in the training set (Antol et al., 2015).) of the task further prove this point: Using only the questions, humans were able to get 40% accuracy overall and 67% accuracy on yes/no questions which clearly shows that the language priors play a significant role. In

Table 4.5: Human baselines in VQA-v1 dataset.

Information	Accuracy (%)	Accuracy (%) of yes/no questions
Question	40.81	67.60
Question + Caption	57.47	78.97
Question + Image	83.30	95.77

Table 4.6: Results comparing the informativeness of the proposed dataset.

Model	Wikimedia Captions				Wikipedia Articles			
	Accuracy	Precision	Recall	F1	Accuracy	Precision	Recall	F1
Random	0.5171	0.5171	0.5171	0.5171	0.5255	0.5255	0.5255	0.5255
DistilBERT	80.91	80.89	80.91	80.83	86.54	86.69	86.54	86.58
	(-2.47+2.28)	(-2.47+2.31)	(-2.47+2.28)	(-2.41+2.36)	(-0.97+0.53)	(-1.08+0.83)	(-0.97+0.53)	(-0.99+0.50)
BERT	82.37	82.35	82.37	82.31	85.60	85.69	85.60	85.45
	(-0.88+1.19)	(-0.96+1.10)	(-0.88+1.19)	(-0.97+1.12)	(-0.91+1.35)	(-0.89+1.24)	(-0.91+1.35)	(-1.07+1.49)

order to overcome this problem, Goyal et al. (2017) come up with the second version of the dataset which has additional samples to balance the biased question/answer pairs. This increased the dataset size to 443K, 214K and 453K pairs (question, image) for train, dev, and test sets respectively. The results reported in this Chapter refer to this new dataset as v2, while they refer to the former as v1.

4.2 Results

The first step of experimentation was to measure the informativeness of the collected dataset. A good way of doing this is to perform concreteness classification tasks on multiple sources. Because obtaining the results for concreteness classification for captions only would not be meaningful without any baseline for comparison, we decided to do the same classification with the regular wikipedia articles in order to show the expressiveness of the captions relative to regular texts. To accomplish this task, the June 2020 version of wikidumps are downloaded which consists of $6,957,578$ articles in total.

To prepare the dataset for a comparison, we search for articles in the Wikipedia dataset about the words in UWA MRC Psycholinguistic dataset. Specifically, each article titled with the corresponding words is retrieved. In order to match the samples with the Wikipedia dataset, captions correspond to the same word are concatenated and the words that do not have a wikipedia article are removed. This leaves 4108 samples in the dataset which is partitioned into train (%70), dev (%10), and test (%20) sets randomly.

Table 4.6 shows the results of DistilBERT and BERT along with the random baselines on the aforementioned datasets. Results show that, although the Wikimedia captions give us worse results than the wikipedia articles, results are not far off, making the wikimedia captions almost as informative as the wikipedia text itself.

Table 4.7: Experimental results of the multi-modal pre-training task.

Model	Accuracy	P	R	F1	F1-abs	F1-conc	P-abs	P-conc	R-abs	R-conc
Bert	0.8116	0.8057	0.8116	0.8069	0.6518	0.8708	0.7076	0.8461	0.6042	0.8971
Resnet	0.7001	0.6472	0.7001	0.6383	0.2144	0.8147	0.4658	0.7227	0.1393	0.9335

Table 4.7 shows the experimental results on the test splits of multi-modal pre-training task. Although the image model and text model is trained with concrete and abstract samples respectively, we show all results belonging to all models.

Several conclusions can be drawn from the results. Firstly, the results comply with Hessel et al. (2018) and Bruni et al. (2014a): identifying concrete concepts is much easier than identifying abstract concepts. Both Resnet and BERT models perform above 0.8 in terms of F1 scores for concrete class, while the F1 of abstract class turns out to be significantly lower with 21.5 and 65.2 respectively. These results show that both image and text models struggle more with abstract concepts compared to concrete concepts.

Secondly, the results of Resnet agrees with the scientific work (i.e. Andrews et al. (2009), Vigliocco et al. (2009)) on human language acquisition and therefore comply with the curriculum learning objectives in this work: experiential information is used early in language acquisition on concrete concepts while leaving its place to textual information for learning abstract concepts. Result show that the image model is capable at learning concrete concepts with high precision and recall numbers (72.3 and 93.3 respectively) while it fails at abstract concepts. This phenomenon is clearer in the case of recall value (13.9) rather than precision (46.6) where the results are somewhat more acceptable. The difference between the two values show that the image model is not sensitive towards abstract samples. It can be argued that, no matter how abstract an idea is, one needs to find a concrete example to show that in an image. For example, the image/caption pairs returned for the search word "dream" frequently contain images of places. Although the word itself can safely be considered as abstract, in order to represent it in the images one needs to find a particular and concrete idea/object to represent it as an image. Therefore, it can be stated that the images almost always contain concrete concepts and to determine abstractness from images, the variance and diversity of images belonging to a particular concept should be used instead of individual images [2].

The proposed model is also tested on the downstream task of Visual Question answering. Results can be seen in Table 4.8. The best result is obtained when both Multi-modal pre-training and attentive pooling mechanism are used, although the performance is consistent across all configurations. In terms of accuracy, there is a 1.01% difference between the best performing model (with Multi-modal pre-training and attentive pooling) and the worst (with Fully connected layer and without multi-modal pre-training). Per-

[2]the variance in images for the word "tomato" is very low, with the first 25 results are all images of single or a couple of red tomatoes, while the variance in images for the word "dream" is very high, ranging from the picture of places, famous people to screenshots of literary work.

Table 4.8: Model performance on VQA dataset. (FC = Fully-connected, AP = Attentive pooling)

Model	Multi-modal Pre-training	Combination Method	Accuracy	F1	P	R
Bert+Resnet	✗	FC	53.12	50.71	54.07	53.12
Bert+Resnet	✓	FC	53.17	52.79	53.34	53.17
Bert+Resnet	✗	AP	53.56	52.91	53.69	53.56
Bert+Resnet	✓	AP	54.13	54.08	54.07	54.13

Table 4.9: Results of the ablation study. Relative performance improvements (%) of each component in terms of F1. (MMPT = Multi-modal pre-training, FC = Fully-connected, AP = Attentive pooling)

	FC	MMPT + FC	AP	MMPT + AP
FC	0	4.10	4.34	6.65
MMPT + FC		0	0.23	2.44
AP			0	2.21
MMPT + AP				0

formance difference becomes more significant in terms of F1: a 3.37% increase can be observed between the best and worst performing models (model with multi-modal pre-training and attentive pooling, and model without multi-modal pre-training and with a fully connected layer respectively, similar to the previous case).

Performance differences can be observed better with the ablation studies. Table 4.9 reports the relative improvements of each component. Each column represents the percentage increase in relative performance when the feature/component in row is replaced or enhanced by the feature/component in the column. It can be seen from the results that multi-modal pre-training increases the performance of the model regardless of the underlying fusion mechanism (Fully-connected or attentive pooling). It leads to 4.1% increase when it is used with fully-connected layers and leads to 2.21% increase when it is used with attentive pooling networks. Similarly, we can see that the attentive pooling mechanism improves the performance of the model in both cases: when the fully-connected layer is replaced with attentive pooling, it amount to an increase of 4.34% when there is multi-modal pre-training and to an increase of 2.44% when there is no multi-modal pre-training. Additionally, this shows that the benefit of using attentive pooling mechanism is somewhat more than the benefit of using multi-modal pre-training. Overall, as the results suggest, using both attentive pooling and multi-modal pre-training are proved to be useful and lead to an increase in performance up to 6.65% compared to the baseline model.

It can be argued that the performance increase obtained by applying a curriculum learning scheme could also be a by-product of additional data that is introduced to the model via Wikimedia Commons dataset. To show whether the performance increase comes from the additional data or the curriculum learning methodology itself, we conducted an additional experiment where extra pre-training is performed without any curriculum

Table 4.10: Effect of Curriculum Learning on the proposed model on VQA.

Multi-modal Pre-training	Accuracy	F1	P	R
No additional pre-training	53.56	52.91	53.69	53.56
Additional pre-training	53.68	53.61	53.68	53.68
Additional pre-training w/ Curriculum Learning	54.13	54.08	54.07	54.13

structure. Instead, the model is trained with Wikimedia Commons data in random order. The results can be seen in Table 4.10 where the results from performing no additional pre-training, performing additional training with no curriculum learning and performing the curriculum learning are compared. Although the experiments show that the additional data can explain some of the performance gain, it cannot be accounted for all. The results show that the model trained with a curriculum learning approach still outperforms the model trained with randomly ordered training set.

Table 4.11 shows the performance of the models described in Section 2.10.2 on the VQA task. We share the results on version 1 and version 2, though it would only be fair to compare the models that run on the same version. Models that run on both versions (Stacked attention network (SAN) and GVQA) suggest that a performance difference between $3 - 7\%$ can be expected between the versions, most likely due to the effect of language priors mentioned in the previous sections. Human baselines, obtained on the 3k samples in the training set of v1 dataset, are also provided.

Although human baselines are on v1 and our performance is on the v2 version of the dataset, our 54.13% accuracy indicates that the model can perform similar to humans when given only questions and corresponding captions without images. Compared to the other models, ours' performed better than the earlier models but can not reach the success obtained by the state-of-the-art model (VilBERT), which has 70.92% accuracy. VilBERT processes paired visiolinguistic data in the architecture of BERT to exploit visual grounding in a task-agnostic way.

Table 4.11: Experimental results on VQA task. Top part shows human baselines.

Model	Dataset Version	Accuracy
Question	v1	40.81
Question + Caption	v1	57.47
Question + Image	v1	83.30
SAN (Yang et al., 2016)	v1	58.9
GVQA (Agrawal et al., 2018)	v1	51.12
SAN (Yang et al., 2016)	v2	52.2
GVQA (Agrawal et al., 2018)	v2	48.24
Anderson et al. (2018)	v2	70.34
DFAF (Gao et al., 2019)	v2	70.34
VilBERT (Lu et al., 2019)	v2	70.92
ours	v2	54.13

It should be noted that there are subtle but vital differences between our model and the VilBERT model. The main focus of VilBERT is to process text and image streams in parallel under the transformer architecture to encode their relationship in a pre-trained model to have optimized performance in downstream tasks. On the other hand, the main focus of this work is to optimize the model for the fusion of modalities and curriculum learning. Although our work is much similar to earlier multi-modal works in this regard, our model is a language pre-training model, not a task-specific architecture. The main difference in our work is to add curriculum learning methodology on top of the pre-trained models.

Other than the main focus described above, several reasons might lead to the performance discrepancy between the proposed model and the state-of-the-art models, such as VilBERT. First, the number of learnable parameters in VilBERT is much greater than the proposed model (~600M vs. ~170M). Second, VilBERT uses the Faster-RCNN Ren et al. (2015) model to match each word in the text with the corresponding image patch, while our model uses the Resnet-152 model on the entire image. One could argue that the better alignment provided by the faster-RCNN method might lead to better learning since the model also learns which part in the image a particular word corresponds to. Providing such an alignment could also benefit the proposed model for catching up with the performance of the state-of-the-art models.

CHAPTER 5

CONCLUSION & FUTURE WORK

The aim of this study is to provide a contribution to one of the oldest and most predominant subjects in computer science: language modeling. Since the distributional hypothesis was formed in early 1950's, many models, which use many different architectures and methodologies, have been introduced in this field. All of these aforementioned models focused on a single modality where a language learner is trained with plain text. Lately, however, the focus is shifted from single modal language models to multi-modal language models. Increase in the success of neural models, cheaper and more powerful hardware sources and the advances in the cognitive science were the major driving forces behind this change.

Similar to this latest trend, the goal of this work is to create a language model/ representation technique inspired from the advances in cognitive science which states that the language acquisition in humans start with the experiential information for concrete concepts and continue with distributional information for abstract concepts. To this end, a combination of BERT and Resnet models combined with attentive pooling mechanism is proposed to construct multi-modal language model and embeddings, in addition to a new dataset composed of image caption pairs from Wikimedia Commons. Image model is trained with the concrete samples from Wikimedia samples first, then the text model is trained with abstract samples in a curriculum learning fashion.

Contribution of this work is two-fold: First, a new dataset, created from Wikimedia Commons, is introduced which has approximately $3.2M$ images, with $630k$ captions, $1.96M$ descriptions, and concreteness labels. Second, a new training scheme, multi-modal pre-training, is introduced. This new learning scheme is inspired from the curriculum learning approaches in artificial intelligence. The results show that, although the model could not outperform state-of-the-art results, the multi-modal pre-training objective can increase the performance of the models significantly. Our results also confirm the findings in the literature by showing that it is harder to detect and classify abstract samples.

There are several arguments that can be made for the improvement of the proposed model. Firstly, a safe bound can be used to differentiate concrete words from the abstract words in the UWA MRC Psycholinguistic dataset . In its current stance, words that have a concreteness rating just above the 400 threshold and words that have a concreteness rating just below the 400 are categorized as concrete and abstract respectively. However,

it can be argued that the differences between such words are negligible, therefore making the samples obtained from them obscure and noisy. Discarding the words between the concreteness rating of, i.e. $300-500$, could help the dataset to reflect the nature of abstract and concrete concepts better.

Second improvement may come from the usage of better image-text alignment models. Although the model goes through each word individually with the text model, the same feature space, constructed by the image model, is used for each word. As state-of-the-art models suggest, better alignment can be provided if one can use an image model that can match each word individually such as an object detection or object segmentation method. In such models, image patches can be aligned with individual words to provide an improved image-text alignment.

REFERENCES

Agrawal, A., D. Batra, D. Parikh, and A. Kembhavi (2018, June). Don't just assume; look and answer: Overcoming priors for visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Akbik, A., D. Blythe, and R. Vollgraf (2018). Contextual string embeddings for sequence labeling. In *COLING 2018, 27th International Conference on Computational Linguistics*, pp. 1638–1649.

Alexandrescu, A. and K. Kirchhoff (2006). Factored neural language models. In *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers*, NAACL-Short '06, Stroudsburg, PA, USA, pp. 1–4. Association for Computational Linguistics.

Anastasopoulos, A., S. Kumar, and H. Liao (2019). Neural language modeling with visual features. arXiv:1903.02930.

Anderson, P., X. He, C. Buehler, D. Teney, M. Johnson, S. Gould, and L. Zhang (2018). Bottom-up and top-down attention for image captioning and visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 6077–6086.

Andrews, M., G. Vigliocco, and D. Vinson (2009). Integrating experiential and distributional data to learn semantic representations. *Psychological Review 116*(3), 463–498.

Antol, S., A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. L. Zitnick, and D. Parikh (2015). VQA: Visual Question Answering. In *International Conference on Computer Vision (ICCV)*.

Bahdanau, D., K. Cho, and Y. Bengio (2014). Neural machine translation by jointly learning to align and translate. *CoRR abs/1409.0473*.

Bakarov, A. (2018). A survey of word embeddings evaluation methods. *CoRR abs/1801.09536*.

Baker, C. F., C. J. Fillmore, and J. B. Lowe (1998). The berkeley framenet project. In *Proceedings of the 17th International Conference on Computational Linguistics - Volume 1*, COLING '98, Stroudsburg, PA, USA, pp. 86–90. Association for Computational Linguistics.

Bengio, Y., R. Ducharme, P. Vincent, and C. Janvin (2003, March). A neural probabilistic language model. *J. Mach. Learn. Res. 3*, 1137–1155.

Bengio, Y., J. Louradour, R. Collobert, and J. Weston (2009). Curriculum learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, New York, NY, USA, pp. 41–48. Association for Computing Machinery.

Bentivogli, L., I. Dagan, H. T. Dang, D. Giampiccolo, and B. Magnini (2009). The fifth pascal recognizing textual entailment challenge. In *In Proc Text Analysis Conference (TAC'09*.

Bian, J., B. Gao, and T.-Y. Liu (2014). Knowledge-powered deep learning for word embedding. In *Proceedings of the 2014th European Conference on Machine Learning and Knowledge Discovery in Databases - Volume Part I*, ECMLPKDD'14, Berlin, Heidelberg, pp. 132–148. Springer-Verlag.

Bishop, C. M. (1995). *Neural Networks for Pattern Recognition*. USA: Oxford University Press, Inc.

Bisk, Y., R. Zellers, R. L. Bras, J. Gao, and Y. Choi (2019). PIQA: reasoning about physical commonsense in natural language. *CoRR abs/1911.11641*.

Blei, D. M., A. Y. Ng, and M. I. Jordan (2003, March). Latent dirichlet allocation. *J. Mach. Learn. Res. 3*, 993–1022.

Bojanowski, P., E. Grave, A. Joulin, and T. Mikolov (2016). Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics 5*, 135–146.

Bollacker, K., C. Evans, P. Paritosh, T. Sturge, and J. Taylor (2008). Freebase: A collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*, SIGMOD '08, New York, NY, USA, pp. 1247–1250. ACM.

Borin, L., M. Forsberg, and L. Lönngren (2013). Saldo: a touch of yin to wordnet's yang. *Language Resources and Evaluation 47*(4), 1191–1211.

Botha, J. A. and P. Blunsom (2014). Compositional morphology for word representations and language modelling. In *Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32*, ICML'14, pp. II–1899–II–1907. JMLR.org.

Brants, S., S. Dipper, P. Eisenberg, S. Hansen-Schirra, E. König, W. Lezius, C. Rohrer, G. Smith, and H. Uszkoreit (2004, Dec). Tiger: Linguistic interpretation of a german corpus. *Research on Language and Computation 2*(4), 597–620.

Braun, S., D. Neil, and S.-C. Liu (2017). A curriculum learning method for improved noise robustness in automatic speech recognition. *2017 25th European Signal Processing Conference (EUSIPCO)*, 548–552.

Brown, T. B., B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei (2020). Language models are few-shot learners. *CoRR abs/2005.14165*.

Bruni, E., G. Boleda, M. Baroni, and N.-K. Tran (2012, July). Distributional semantics in technicolor. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Jeju Island, Korea, pp. 136–145. Association for Computational Linguistics.

Bruni, E., N. K. Tran, and M. Baroni (2014a, January). Multimodal distributional semantics. *J. Artif. Int. Res. 49*(1), 1–47.

Bruni, E., N. K. Tran, and M. Baroni (2014b, January). Multimodal distributional semantics. *J. Artif. Int. Res. 49*(1), 1–47.

Camacho-Collados, J. and M. T. Pilehvar (2018, September). From word to sense embeddings: A survey on vector representations of meaning. *J. Artif. Int. Res. 63*(1), 743–788.

Cao, K. and M. Rei (2016, August). A joint model for word embedding and word morphology. In *Proceedings of the 1st Workshop on Representation Learning for NLP*, Berlin, Germany, pp. 18–26. Association for Computational Linguistics.

Caubrière, A., N. A. Tomashenko, A. Laurent, E. Morin, N. Camelin, and Y. Estève (2019). Curriculum-based transfer learning for an effective end-to-end spoken language understanding and domain portability. *CoRR abs/1906.07601*.

Cer, D., M. Diab, E. Agirre, I. Lopez-Gazpio, and L. Specia (2017, August). SemEval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, Vancouver, Canada, pp. 1–14. Association for Computational Linguistics.

Chang, H.-S., E. Learned-Miller, and A. McCallum (2017). Active bias: Training more accurate neural networks by emphasizing high variance samples. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, Red Hook, NY, USA, pp. 1003–1013. Curran Associates Inc.

Changpinyo, S., W. Chao, B. Gong, and F. Sha (2016). Synthesized classifiers for zero-shot learning. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5327–5336.

Chen, S. F. and J. Goodman (1996). An empirical study of smoothing techniques for language modeling. In *Proceedings of the 34th Annual Meeting on Association for Computational Linguistics*, ACL '96, pp. 310–318. Association for Computational Linguistics.

Chen, T., R. Xu, Y. He, and X. Wang (2015). Improving distributed representation of word sense via wordnet gloss composition and context clustering. In *ACL*.

Chen, X., Z. Liu, and M. Sun (2014, October). A unified model for word sense representation and disambiguation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Doha, Qatar, pp. 1025–1035. Association for Computational Linguistics.

chen, Z., H. Zhang, X. Zhang, and L. Zhao (2018). Quora question pairs.

Cheng, J. and D. Kartsaklis (2015, September). Syntax-aware multi-sense word embeddings for deep compositional models of meaning. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, Lisbon, Portugal, pp. 1531–1542. Association for Computational Linguistics.

Cho, K., B. van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio (2014, October). Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Doha, Qatar, pp. 1724–1734. Association for Computational Linguistics.

Coates, A. and A. Y. Ng (2011). The importance of encoding versus training with sparse coding and vector quantization. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*, ICML'11, Madison, WI, USA, pp. 921–928. Omnipress.

Collell Talleda, G., T. Zhang, and M.-F. Moens (2017). Imagined visual representations as multimodal embeddings. pp. 4378–4384. Singh, Satinder P: AAAI.

Collobert, R. and J. Weston (2008). A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, New York, NY, USA, pp. 160–167.

Collobert, R., J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa (2011, November). Natural language processing (almost) from scratch. *J. Mach. Learn. Res. 12*, 2493–2537.

Coltheart, M. (1981). The mrc psycholinguistic database. *The Quarterly Journal of Experimental Psychology Section A 33*(4), 497–505.

Cotterell, R. and H. Schütze (2015, May–June). Morphological word-embeddings. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Denver, Colorado, pp. 1287–1292. Association for Computational Linguistics.

Cotterell, R., H. Schütze, and J. Eisner (2016). Morphological smoothing and extrapolation of word embeddings. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1651–1660. Association for Computational Linguistics.

Creutz, M. and K. Lagus (2007, February). Unsupervised models for morpheme segmentation and morphology learning. *ACM Trans. Speech Lang. Process. 4*(1), 3:1–3:34.

Cui, Q., B. Gao, J. Bian, S. Qiu, H. Dai, and T.-Y. Liu (2015, August). Knet: A general framework for learning word embedding using morphological knowledge. *ACM Trans. Inf. Syst. 34*(1), 4:1–4:25.

Dai, A. M. and Q. V. Le (2015). Semi-supervised sequence learning. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'15, Cambridge, MA, USA, pp. 3079–3087. MIT Press.

Dai, Z., Z. Yang, Y. Yang, J. Carbonell, Q. Le, and R. Salakhutdinov (2019, July). Transformer-XL: Attentive language models beyond a fixed-length context. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, Florence, Italy, pp. 2978–2988. Association for Computational Linguistics.

Deerwester, S., S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman (1990). Indexing by latent semantic analysis. *JOURNAL OF THE AMERICAN SOCIETY FOR INFORMATION SCIENCE 41*(6), 391–407.

Deng, J., W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei (2009). Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee.

Devereux, B. J., L. K. Tyler, J. Geertzen, and B. Randall (2014). The centre for speech, language and the brain (cslb) concept property norms. *Behavior research methods 46*(4), 1119–1127.

Devlin, J., M. Chang, K. Lee, and K. Toutanova (2019). BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pp. 4171–4186.

Dolan, B., C. Quirk, and C. Brockett (2004, aug 23–aug 27). Unsupervised construction of large paraphrase corpora: Exploiting massively parallel news sources. In *COLING 2004: Proceedings of the 20th International Conference on Computational Linguistics*, Geneva, Switzerland, pp. 350–356. COLING.

Dos Santos, C. N. and B. Zadrozny (2014). Learning character-level representations for part-of-speech tagging. In *Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32*, ICML'14, pp. II–1818–II–1826. JMLR.org.

Eisenschtat, A. and L. Wolf (2017). Linking image and text with 2-way nets. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1855–1865.

Elman, J. L. (1990). Finding structure in time. *Cognitive Science 14*(2), 179–211.

Elman, J. L. (1993). Learning and development in neural networks: the importance of starting small. *Cognition 48*(1), 71–99.

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

Feng, Y. and M. Lapata (2010). Visual information in semantic representation. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, HLT '10, USA, pp. 91–99. Association for Computational Linguistics.

Finkelstein, L., E. Gabrilovich, Y. Matias, E. Rivlin, Z. Solan, G. Wolfman, and E. Ruppin (2001). Placing search in context: The concept revisited. In *Proceedings of the 10th International Conference on World Wide Web*, WWW '01, New York, NY, USA, pp. 406–414. ACM.

Frome, A., G. S. Corrado, J. Shlens, S. Bengio, J. Dean, and T. Mikolov (2013). Devise: A deep visual-semantic embedding model. In *In NIPS*.

Gao, P., Z. Jiang, H. You, P. Lu, S. C. H. Hoi, X. Wang, and H. Li (2019). Dynamic fusion with intra- and inter-modality attention flow for visual question answering. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6632–6641.

Gilhooly, K. J. and R. H. Logie (1980). Age-of-acquisition, imagery, concreteness, familiarity, and ambiguity measures for 1,944 words. *Behavior research methods & instrumentation 12*(4), 395–427.

Gong, C. (2017). Exploring commonality and individuality for multi-modal curriculum learning. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, AAAI'17, pp. 1926–1933. AAAI Press.

Gong, C., D. Tao, S. J. Maybank, W. Liu, G. Kang, and J. Yang (2016). Multi-modal curriculum learning for semi-supervised image classification. *IEEE Transactions on Image Processing 25*(7), 3249–3260.

Gong, Y., H. Luo, and J. Zhang (2017). Natural language inference over interaction space. *CoRR abs/1709.04348*.

Goyal, Y., T. Khot, D. Summers-Stay, D. Batra, and D. Parikh (2017). Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6325–6334.

Graves, A. and J. Schmidhuber (2005). Framewise phoneme classification with bidirectional lstm and other neural network architectures. *Neural networks : the official journal of the International Neural Network Society 18 5-6*, 602–10.

Griffiths, T. L., J. B. Tenenbaum, and M. Steyvers (2007). Topics in semantic representation. *Psychological Review 114*, 2007.

Guo, J., W. Che, H. Wang, and T. Liu (2014, August). Learning sense-specific word embeddings by exploiting bilingual resources. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, Dublin, Ireland, pp. 497–507. Dublin City University and Association for Computational Linguistics.

Gutmann, M. U. and A. Hyvärinen (2012, February). Noise-contrastive estimation of unnormalized statistical models, with applications to natural image statistics. *J. Mach. Learn. Res. 13*(1), 307–361.

Hacohen, G. and D. Weinshall (2019, 09–15 Jun). On the power of curriculum learning in training deep networks. In K. Chaudhuri and R. Salakhutdinov (Eds.), *Proceedings*

of the 36th International Conference on Machine Learning, Volume 97 of *Proceedings of Machine Learning Research*, pp. 2535–2544. PMLR.

Harris, Z. S. (1954). Distributional structure. *Word 10*(2-3), 146–162.

He, K., X. Zhang, S. Ren, and J. Sun (2016). Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778.

He, M., Y. Song, K. Xu, and Y. Dong (2020). On the role of conceptualization in commonsense knowledge graph construction. *ArXiv abs/2003.03239*.

Hessel, J., D. Mimno, and L. Lee (2018, June). Quantifying the visual concreteness of words and topics in multimodal datasets. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, New Orleans, Louisiana, pp. 2194–2205. Association for Computational Linguistics.

Hill, F. and A. Korhonen (2014, October). Learning abstract concept embeddings from multi-modal data: Since you probably can't see what I mean. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Doha, Qatar, pp. 255–265. Association for Computational Linguistics.

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

Hinton, G. E., J. L. McClelland, and D. E. Rumelhart (1986). Parallel distributed processing: Explorations in the microstructure of cognition, vol. 1. Chapter Distributed Representations, pp. 77–109. Cambridge, MA, USA: MIT Press.

Hochreiter, S. and J. Schmidhuber (1997, November). Long short-term memory. *Neural Comput. 9*(8), 1735–1780.

Howard, J. and S. Ruder (2018, July). Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Melbourne, Australia, pp. 328–339. Association for Computational Linguistics.

Huang, E. H., R. Socher, C. D. Manning, and A. Y. Ng (2012). Improving word representations via global context and multiple word prototypes. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers - Volume 1*, ACL '12, Stroudsburg, PA, USA, pp. 873–882. Association for Computational Linguistics.

Iacobacci, I. and R. Navigli (2019). Lstmembed: Learning word and sense representations from a large semantically annotated corpus with long short-term memories. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pp. 1685–1695.

Iacobacci, I., M. T. Pilehvar, and R. Navigli (2015). Sensembed: Learning sense embeddings for word and relational similarity. 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)*, pp. 95–105. Association for Computational Linguistics.

Jarmasz, M. (2003). Roget's thesaurus as a lexical resource for natural language processing. *ArXiv abs/1204.0140*.

Jauhar, S. K., C. Dyer, and E. Hovy (2015, May–June). Ontologically grounded multi-sense representation learning for semantic vector space models. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Denver, Colorado, pp. 683–693. Association for Computational Linguistics.

Jiang, L., D. Meng, Q. Zhao, S. Shan, and A. G. Hauptmann (2015). Self-paced curriculum learning. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, AAAI'15, pp. 2694–2700. AAAI Press.

Joshi, M., E. Choi, D. S. Weld, and L. Zettlemoyer (2017). Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. In R. Barzilay and M. Kan (Eds.), *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pp. 1601–1611. Association for Computational Linguistics.

Karpathy, A. and L. Fei-Fei (2017). Deep visual-semantic alignments for generating image descriptions. *IEEE Transactions on Pattern Analysis and Machine Intelligence 39*(4), 664–676.

Karpathy, A., A. Joulin, and L. Fei-Fei (2014). Deep fragment embeddings for bidirectional image sentence mapping. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, Cambridge, MA, USA, pp. 1889–1897. MIT Press.

Kiela, D., L. Rimell, I. Vulić, and S. Clark (2015, July). Exploiting image generality for lexical entailment detection. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference*

on Natural Language Processing (Volume 2: Short Papers), Beijing, China, pp. 119–124. Association for Computational Linguistics.

Kim, J., M. Ma, K. Kim, S. Kim, and C. Yoo (2019). Gaining extra supervision via multi-task learning for multi-modal video question answering. *2019 International Joint Conference on Neural Networks (IJCNN)*, 1–8.

Kim, T.-H. and J. Choi (2018). Screenernet: Learning self-paced curriculum for deep neural networks. *arXiv: Computer Vision and Pattern Recognition*.

Kim, Y., Y. Jernite, D. A. Sontag, and A. M. Rush (2016). Character-aware neural language models. In D. Schuurmans and M. P. Wellman (Eds.), *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA*, pp. 2741–2749. AAAI Press.

Kiros, R., R. Salakhutdinov, and R. Zemel (2014a). Multimodal neural language models. In *Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32*, ICML'14, pp. II–595–II–603. JMLR.org.

Kiros, R., R. Salakhutdinov, and R. Zemel (2014b). Unifying visual-semantic embeddings with multimodal neural language models. *ArXiv abs/1411.2539*.

Klein, B., G. Lev, G. Sadeh, and L. Wolf (2015). Associating neural word embeddings with deep image representations using fisher vectors. In *In CVPR*.

Kneser, R. and H. Ney (1995, May). Improved backing-off for m-gram language modeling. In *1995 International Conference on Acoustics, Speech, and Signal Processing*, Volume 1, pp. 181–184 vol.1.

Kodirov, E., T. Xiang, and S. Gong (2017). Semantic autoencoder for zero-shot learning. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4447–4456.

Krizhevsky, A., I. Sutskever, and G. E. Hinton (2012). Imagenet classification with deep convolutional neural networks. In *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'12, Red Hook, NY, USA, pp. 1097–1105. Curran Associates Inc.

Kumar, M., B. Packer, and D. Koller (2010). Self-paced learning for latent variable models. In J. Lafferty, C. Williams, J. Shawe-Taylor, R. Zemel, and A. Culotta (Eds.), *Advances in Neural Information Processing Systems*, Volume 23. Curran Associates, Inc.

Kumar, S., S. Jat, K. Saxena, and P. Talukdar (2019). Zero-shot word sense disambiguation using sense definition embeddings. In *Proceedings of the 57th Conference of the*

Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers, pp. 5670–5681.

Lai, G., Q. Xie, H. Liu, Y. Yang, and E. Hovy (2017, September). RACE: Large-scale ReAding comprehension dataset from examinations. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, Copenhagen, Denmark, pp. 785–794. Association for Computational Linguistics.

Lan, Z., M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut (2020). Albert: A lite bert for self-supervised learning of language representations. In *International Conference on Learning Representations*.

Landauer, T. K. and S. T. Dutnais (1997). A solution to plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *PSYCHOLOGICAL REVIEW 104*(2), 211–240.

Lee, K.-H., X. Chen, G. Hua, H. Hu, and X. He (2018). Stacked cross attention for image-text matching. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 201–216.

Lee, Y. J. and K. Grauman (2011). Learning the easy things first: Self-paced visual category discovery. In *CVPR 2011*, pp. 1721–1728.

Leong, C. W. and R. Mihalcea (2011, November). Going beyond text: A hybrid image-text approach for measuring word relatedness. In *Proceedings of 5th International Joint Conference on Natural Language Processing*, Chiang Mai, Thailand, pp. 1403–1407. Asian Federation of Natural Language Processing.

Levesque, H., E. Davis, and L. Morgenstern (2012). The winograd schema challenge. In *Thirteenth International Conference on the Principles of Knowledge Representation and Reasoning*.

Levy, O. and Y. Goldberg (2014). Dependency-based word embeddings. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 302–308. Association for Computational Linguistics.

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

Li, J. and D. Jurafsky (2015, September). Do multi-sense embeddings improve natural language understanding? In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, Lisbon, Portugal, pp. 1722–1732. Association for Computational Linguistics.

Liang, F. M. (1983a). Word hy-phen-a-tion by com-put-er. Technical report.

Liang, F. M. (1983b). *Word Hy-phen-a-tion by Com-put-er (Hyphenation, Computer)*. Ph. D. thesis, Stanford University, Stanford, CA, USA. AAI8329742.

Ling, W., C. Dyer, A. W. Black, I. Trancoso, R. Fermandez, S. Amir, L. Marujo, and T. Luís (2015, September). Finding function in form: Compositional character models for open vocabulary word representation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, Lisbon, Portugal, pp. 1520–1530. Association for Computational Linguistics.

Ling, W., Y. Tsvetkov, S. Amir, R. Fermandez, C. Dyer, A. W. Black, I. Trancoso, and C.-C. Lin (2015, September). Not all contexts are created equal: Better word representations with variable attention. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, Lisbon, Portugal, pp. 1367–1372. Association for Computational Linguistics.

Liu, Q., H. Jiang, S. Wei, Z.-H. Ling, and Y. Hu (2015, July). Learning semantic word embeddings based on ordinal knowledge constraints. 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)*, Beijing, China, pp. 1501–1511. Association for Computational Linguistics.

Liu, Q., M. J. Kusner, and P. Blunsom (2020). A survey on contextual embeddings. *ArXiv abs/2003.07278*.

Liu, Y., Y. Guo, E. M. Bakker, and M. S. Lew (2017). Learning a recurrent residual fusion network for multimodal matching. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 4127–4136.

Liu, Y., Z. Liu, T.-S. Chua, and M. Sun (2015). Topical word embeddings. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, AAAI'15, pp. 2418–2424. AAAI Press.

Liu, Y., M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov (2019). Roberta: A robustly optimized bert pretraining approach. *ArXiv abs/1907.11692*.

Lotfian, R. and C. Busso (2019, April). Curriculum learning for speech emotion recognition from crowdsourced labels. *IEEE/ACM Trans. Audio, Speech and Lang. Proc. 27*(4), 815–826.

Lowe, D. (1999). Object recognition from local scale-invariant features. In *Proceedings of the Seventh IEEE International Conference on Computer Vision*, Volume 2, pp. 1150–1157 vol.2.

Lu, J., D. Batra, D. Parikh, and S. Lee (2019). Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (Eds.), *Advances in Neural Information Processing Systems*, Volume 32. Curran Associates, Inc.

Luong, T., R. Socher, and C. Manning (2013, August). Better word representations with recursive neural networks for morphology. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*, Sofia, Bulgaria, pp. 104–113. Association for Computational Linguistics.

Luu, A. T., Y. Tay, S. C. Hui, and S. K. Ng (2016, November). Learning term embeddings for taxonomic relation identification using dynamic weighting neural network. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, Austin, Texas, pp. 403–413. Association for Computational Linguistics.

Ma, Y., X. Xu, F. Shen, and H. T. Shen (2020). Similarity preserving feature generating networks for zero-shot learning. *Neurocomputing 406*, 333–342.

McCann, B., J. Bradbury, C. Xiong, and R. Socher (2017). Learned in translation: Contextualized word vectors. In I. Guyon, U. von Luxburg, S. Bengio, H. M. Wallach, R. Fergus, S. V. N. Vishwanathan, and R. Garnett (Eds.), *NIPS*, pp. 6297–6308.

Melamud, O., J. Goldberger, and I. Dagan (2016). context2vec: Learning generic context embedding with bidirectional LSTM. In *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning, CoNLL 2016, Berlin, Germany, August 11-12, 2016*, pp. 51–61.

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

Mikolov, T., M. Karafiát, L. Burget, J. Cernocký, and S. Khudanpur (2010). Recurrent neural network based language model. In *INTERSPEECH 2010, 11th Annual Conference of the International Speech Communication Association, Makuhari, Chiba, Japan, September 26-30, 2010*, pp. 1045–1048.

Mikolov, T., I. Sutskever, K. Chen, G. Corrado, and J. Dean (2013). Distributed representations of words and phrases and their compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'13, USA, pp. 3111–3119. Curran Associates Inc.

Miller, G. A. (1995, November). Wordnet: A lexical database for english. *Commun. ACM 38*(11), 39–41.

Miller, G. A. and W. G. Charles (1991). Contextual correlates of semantic similarity. *Language and Cognitive Processes 6*(1), 1–28.

Mnih, A. and G. Hinton (2007). Three new graphical models for statistical language modelling. In *Proceedings of the 24th International Conference on Machine Learning*, ICML '07, pp. 641–648.

Mnih, A. and G. Hinton (2008). A scalable hierarchical distributed language model. In *Proceedings of the 21st International Conference on Neural Information Processing Systems*, NIPS'08, USA, pp. 1081–1088. Curran Associates Inc.

Mnih, A. and K. Kavukcuoglu (2013). Learning word embeddings efficiently with noise-contrastive estimation. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'13, USA, pp. 2265–2273. Curran Associates Inc.

Mnih, A. and Y. W. Teh (2012). A fast and simple algorithm for training neural probabilistic language models. In *Proceedings of the 29th International Coference on International Conference on Machine Learning*, ICML'12, USA, pp. 419–426. Omnipress.

Morerio, P., J. Cavazza, R. Volpi, R. Vidal, and V. Murino (2017). Curriculum dropout. *2017 IEEE International Conference on Computer Vision (ICCV)*, 3564–3572.

Morin, F. and Y. Bengio (2005). Hierarchical probabilistic neural network language model. In R. G. Cowell and Z. Ghahramani (Eds.), *Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics*, pp. 246–252. Society for Artificial Intelligence and Statistics.

Mrkšić, N., D. Ó Séaghdha, B. Thomson, M. Gašić, L. M. Rojas-Barahona, P.-H. Su, D. Vandyke, T.-H. Wen, and S. Young (2016, June). Counter-fitting word vectors to linguistic constraints. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, San Diego, California, pp. 142–148. Association for Computational Linguistics.

Navigli, R. (2009, February). Word sense disambiguation: A survey. *ACM Comput. Surv. 41*(2).

Navigli, R. and S. P. Ponzetto (2012, December). Babelnet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *Artif. Intell. 193*, 217–250.

Neelakantan, A., J. Shankar, A. Passos, and A. McCallum (2014, October). Efficient non-parametric estimation of multiple embeddings per word in vector space. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Doha, Qatar, pp. 1059–1069. Association for Computational Linguistics.

Newport, E. L. (1990). Maturational constraints on language learning. *Cognitive Science 14*(1), 11–28.

Nguyen, D. Q., D. Q. Nguyen, A. Modi, S. Thater, and M. Pinkal (2017, August). A mixture model for learning multi-sense word embeddings. In *Proceedings of the 6th Joint Conference on Lexical and Computational Semantics (*SEM 2017)*, Vancouver, Canada, pp. 121–127. Association for Computational Linguistics.

Nguyen, K. A., M. Köper, S. Schulte im Walde, and N. T. Vu (2017, September). Hierarchical embeddings for hypernymy detection and directionality. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, Copenhagen, Denmark, pp. 233–243. Association for Computational Linguistics.

Nguyen, K. A., S. Schulte im Walde, and N. T. Vu (2016, August). Integrating distributional lexical contrast into word embeddings for antonym-synonym distinction. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, Berlin, Germany, pp. 454–459. Association for Computational Linguistics.

Nieto Piña, L. and R. Johansson (2015, September). A simple and efficient method to generate word sense representations. In *Proceedings of the International Conference Recent Advances in Natural Language Processing*, Hissar, Bulgaria, pp. 465–472. INCOMA Ltd. Shoumen, BULGARIA.

Norouzi, M., T. Mikolov, S. Bengio, Y. Singer, J. Shlens, A. Frome, G. Corrado, and J. Dean (2014). Zero-shot learning by convex combination of semantic embeddings. In *International Conference on Learning Representations*.

Ororbia, A., A. Mali, M. Kelly, and D. Reitter (2019, July). Like a baby: Visually situated neural language acquisition. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, Florence, Italy, pp. 5127–5136. Association for Computational Linguistics.

Paivio, A., J. C. Yuille, and S. A. Madigan (1968). Concreteness, imagery, and meaningfulness values for 925 nouns. *Journal of experimental psychology 76*(1p2), 1.

Paperno, D., G. Kruszewski, A. Lazaridou, Q. N. Pham, R. Bernardi, S. Pezzelle, M. Baroni, G. Boleda, and R. Fernández (2016). The LAMBADA dataset: Word prediction

requiring a broad discourse context. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*. The Association for Computer Linguistics.

Pavlick, E., P. Rastogi, J. Ganitkevitch, B. Van Durme, and C. Callison-Burch (2015, July). PPDB 2.0: Better paraphrase ranking, fine-grained entailment relations, word embeddings, and style classification. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, Beijing, China, pp. 425–430. Association for Computational Linguistics.

Pelevina, M., N. Arefiev, C. Biemann, and A. Panchenko (2016). Making sense of word embeddings. In *Proceedings of the 1st Workshop on Representation Learning for NLP*, pp. 174–183. Association for Computational Linguistics.

Pennington, J., R. Socher, and C. D. Manning (2014). Glove: Global vectors for word representation. In *EMNLP*, pp. 1532–1543. ACL.

Peters, M. E., M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer (2018). Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, pp. 2227–2237.

Qiu, S., Q. Cui, J. Bian, B. Gao, and T.-Y. Liu (2014, August). Co-learning of word representations and morpheme representations. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, Dublin, Ireland, pp. 141–150. Dublin City University and Association for Computational Linguistics.

Rajpurkar, P., R. Jia, and P. Liang (2018a, July). Know what you don't know: Unanswerable questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, Melbourne, Australia, pp. 784–789. Association for Computational Linguistics.

Rajpurkar, P., R. Jia, and P. Liang (2018b, July). Know what you don't know: Unanswerable questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, Melbourne, Australia, pp. 784–789. Association for Computational Linguistics.

Rajpurkar, P., J. Zhang, K. Lopyrev, and P. Liang (2016, November). SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference*

on Empirical Methods in Natural Language Processing, Austin, Texas, pp. 2383–2392. Association for Computational Linguistics.

Reisinger, J. and R. J. Mooney (2010). Multi-prototype vector-space models of word meaning. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, HLT '10, Stroudsburg, PA, USA, pp. 109–117. Association for Computational Linguistics.

Ren, S., K. He, R. Girshick, and J. Sun (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'15, Cambridge, MA, USA, pp. 91–99. MIT Press.

Ripley, B. D. and N. L. Hjort (1995). *Pattern Recognition and Neural Networks* (1st ed.). USA: Cambridge University Press.

Rogers, A., O. Kovaleva, and A. Rumshisky (2020). A primer in bertology: What we know about how bert works. *ArXiv abs/2002.12327*.

Rohrbach, M., M. Stark, and B. Schiele (2011). Evaluating knowledge transfer and zero-shot learning in a large-scale setting. In *CVPR 2011*, pp. 1641–1648.

Romera-Paredes, B. and P. H. S. Torr (2015). An embarrassingly simple approach to zero-shot learning. ICML'15, pp. 2152–2161. JMLR.org.

Rothe, S. and H. Schütze (2015). Autoextend: Extending word embeddings to embeddings for synsets and lexemes. 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)*, pp. 1793–1803. Association for Computational Linguistics.

Rubenstein, H. and J. B. Goodenough (1965, October). Contextual correlates of synonymy. *Commun. ACM 8*(10), 627–633.

Sanh, V., L. Debut, J. Chaumond, and T. Wolf (2019). Distilbert, a distilled version of BERT: smaller, faster, cheaper and lighter. *CoRR abs/1910.01108*.

Santos, C. d., M. Tan, B. Xiang, and B. Zhou (2016). Attentive pooling networks. *arXiv preprint arXiv:1602.03609*.

Schütze, H. (1998). Automatic word sense discrimination. *Computational Linguistics 24*(1), 97–123.

Sezerer, E. and S. Tekir (2021). Incorporating concreteness in multi-modal language models with curriculum learning. *Applied Sciences 11*(17).

Sharma, P., N. Ding, S. Goodman, and R. Soricut (2018, July). Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Melbourne, Australia, pp. 2556–2565. Association for Computational Linguistics.

Shi, H., J. Mao, T. Xiao, Y. Jiang, and J. Sun (2018, August). Learning visually-grounded semantics from contrastive adversarial samples. In *Proceedings of the 27th International Conference on Computational Linguistics*, Santa Fe, New Mexico, USA, pp. 3715–3727. Association for Computational Linguistics.

Shi, Y., M. Larson, and C. M. Jonker (2013). K-component recurrent neural network language models using curriculum learning. In *2013 IEEE Workshop on Automatic Speech Recognition and Understanding*, pp. 1–6.

Simonyan, K. and A. Zisserman (2015). Very deep convolutional networks for large-scale image recognition. *CoRR abs/1409.1556*.

Socher, R. and L. Fei-Fei (2010). Connecting modalities: Semi-supervised segmentation and annotation of images using unaligned text corpora. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 966–973.

Socher, R., M. Ganjoo, C. D. Manning, and A. Y. Ng (2013). Zero-shot learning through cross-modal transfer. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'13, Red Hook, NY, USA, pp. 935–943. Curran Associates Inc.

Socher, R., A. Perelygin, J. Wu, J. Chuang, C. D. Manning, A. Ng, and C. Potts (2013, October). Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, Seattle, Washington, USA, pp. 1631–1642. Association for Computational Linguistics.

Soricut, R. and F. J. Och (2015). Unsupervised morphology induction using word embeddings. In *HLT-NAACL*.

Soviany, P., R. T. Ionescu, P. Rota, and N. Sebe (2021). Curriculum learning: A survey. *CoRR abs/2101.10382*.

Spitkovsky, V. I., H. Alshawi, and D. Jurafsky (2009). Baby steps: How "less is more" in unsupervised dependency parsing. In *NIPS 2009 Workshop on Grammar Induction, Representation of Language and Language Learning*.

Srinivasan, K., K. Raman, J. Chen, M. Bendersky, and M. Najork (2021). Wit: Wikipedia-based image text dataset for multimodal multilingual machine learning. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '21, New York, NY, USA, pp. 2443–2449. Association for Computing Machinery.

Srivastava, R., K. Greff, and J. Schmidhuber (2015a). Highway networks. *ArXiv abs/1505.00387*.

Srivastava, R., K. Greff, and J. Schmidhuber (2015b). Training very deep networks. In *NIPS*.

Sun, Y., S. Wang, Y. Li, S. Feng, X. Chen, H. Zhang, X. Tian, D. Zhu, H. Tian, and H. Wu (2019). Ernie: Enhanced representation through knowledge integration. *ArXiv abs/1904.09223*.

Sun, Y., S. Wang, Y. Li, S. Feng, H. Tian, H. Wu, and H. Wang (2020). Ernie 2.0: A continual pre-training framework for language understanding. *ArXiv abs/1907.12412*.

Szegedy, C., W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich (2015). Going deeper with convolutions. In *Computer Vision and Pattern Recognition (CVPR)*.

Szegedy, C., V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna (2016). Rethinking the inception architecture for computer vision. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2818–2826.

Tang, Y.-P. and S.-J. Huang (2019, Jul.). Self-paced active learning: Query the right thing at the right time. *33*, 5117–5124.

Tekir, S. and Y. Bastanlar (2020). Deep learning: Exemplar studies in natural language processing and computer vision. In *Data Mining-Methods, Applications and Systems*. IntechOpen.

Tian, F., H. Dai, J. Bian, B. Gao, R. Zhang, E. Chen, and T.-Y. Liu (2014, August). A probabilistic model for learning multi-prototype word embeddings. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, Dublin, Ireland, pp. 151–160. Dublin City University and Association for Computational Linguistics.

Tissier, J., C. Gravier, and A. Habrard (2017, September). Dict2vec : Learning word embeddings using lexical dictionaries. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, Copenhagen, Denmark, pp. 254–263. Association for Computational Linguistics.

Turian, J., L. Ratinov, and Y. Bengio (2010). Word representations: A simple and general method for semi-supervised learning. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, ACL '10, Stroudsburg, PA, USA, pp. 384–394. Association for Computational Linguistics.

Turney, P. D. (2001). Mining the web for synonyms: Pmi-ir versus lsa on toefl. In L. De Raedt and P. Flach (Eds.), *Machine Learning: ECML 2001*, Berlin, Heidelberg, pp. 491–502. Springer Berlin Heidelberg.

Turney, P. D. and P. Pantel (2010, January). From frequency to meaning: Vector space models of semantics. *J. Artif. Int. Res. 37*(1), 141–188.

Vaswani, A., N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin (2017). Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Eds.), *Advances in Neural Information Processing Systems 30*, pp. 5998–6008. Curran Associates, Inc.

Vigliocco, G., L. Meteyard, M. Andrews, and S. Kousta (2009). Toward a theory of semantic representation. *Language and Cognition 1*(2), 219–247.

von Ahn, L. and L. Dabbish (2004). Labeling images with a computer game. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '04, New York, NY, USA, pp. 319–326. Association for Computing Machinery.

Vulić, I., N. Mrkšić, R. Reichart, D. Ó Séaghdha, S. Young, and A. Korhonen (2017, July). Morph-fitting: Fine-tuning word vector spaces with simple language-specific rules. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Vancouver, Canada, pp. 56–68. Association for Computational Linguistics.

Wang, A., A. Singh, J. Michael, F. Hill, O. Levy, and S. Bowman (2018, November). GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, Brussels, Belgium, pp. 353–355. Association for Computational Linguistics.

Wang, C., X. He, and A. Zhou (2019). Spherere: Distinguishing lexical relations with hyperspherical relation embeddings. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pp. 1727–1737.

Wang, L., Y. Li, and S. Lazebnik (2016). Learning deep structure-preserving image-text embeddings. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5005–5013.

Wang, W., V. W. Zheng, H. Yu, and C. Miao (2019, January). A survey of zero-shot learning: Settings, methods, and applications. *10*(2).

Wang, X., Y. Chen, and W. Zhu (2021). A survey on curriculum learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1–1.

Warstadt, A., A. Singh, and S. R. Bowman (2019). Neural network acceptability judgments. *Transactions of the Association for Computational Linguistics 7*, 625–641.

Weston, J., S. Bengio, and N. Usunier (2010). Large scale image annotation: Learning to rank with joint word-image embeddings. In *European Conference on Machine Learning*.

Wieting, J., M. Bansal, K. Gimpel, and K. Livescu (2015). From paraphrase database to compositional paraphrase model and back. *Transactions of the Association for Computational Linguistics 3*, 345–358.

Williams, A., N. Nangia, and S. Bowman (2018). A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pp. 1112–1122. Association for Computational Linguistics.

Wittgenstein, L. (1953). *Philosophical Investigations*. Basil Blackwell.

Wu, H., J. Mao, Y. Zhang, Y. Jiang, L. Li, W. Sun, and W.-Y. Ma (2019). Unified visual-semantic embeddings: Bridging vision and language with structured meaning representations. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6602–6611.

Wu, W., H. Li, H. Wang, and K. Q. Zhu (2012). Probase: A probabilistic taxonomy for text understanding. In *Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data*, SIGMOD '12, New York, NY, USA, pp. 481–492. ACM.

Xian, Y., C. H. Lampert, B. Schiele, and Z. Akata (2019). Zero-shot learning—a comprehensive evaluation of the good, the bad and the ugly. *IEEE Transactions on Pattern Analysis and Machine Intelligence 41*(9), 2251–2265.

Xian, Y., T. Lorenz, B. Schiele, and Z. Akata (2018). Feature generating networks for zero-shot learning. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5542–5551.

Xie, G.-S., L. Liu, X. Jin, F. Zhu, Z. Zhang, J. Qin, Y. Yao, and L. Shao (2019, June). Attentive region embedding network for zero-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Xu, B., L. Zhang, Z. Mao, Q. Wang, H. Xie, and Y. Zhang (2020, July). Curriculum learning for natural language understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Online, pp. 6095–6104. Association for Computational Linguistics.

Xu, W., W. Liu, X. Huang, J. Yang, and S. Qiu (2018). Multi-modal self-paced learning for image classification. *Neurocomputing 309*, 134–144.

Xu, W. and A. Rudnicky (2000). Can artificial neural networks learn language models? In *Sixth International Conference on Spoken Language Processing*.

Xu, Y., J. Liu, W. Yang, and L. Huang (2018, July). Incorporating latent meanings of morphological compositions to enhance word embeddings. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Melbourne, Australia, pp. 1232–1242. Association for Computational Linguistics.

Yang, D. and D. M. W. Powers (2005). Measuring semantic similarity in the taxonomy of wordnet. In *Proceedings of the Twenty-eighth Australasian Conference on Computer Science - Volume 38*, ACSC '05, Darlinghurst, Australia, Australia, pp. 315–322. Australian Computer Society, Inc.

Yang, Z., Z. Dai, Y. Yang, J. Carbonell, R. R. Salakhutdinov, and Q. V. Le (2019). Xlnet: Generalized autoregressive pretraining for language understanding. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (Eds.), *Advances in Neural Information Processing Systems 32*, pp. 5753–5763. Curran Associates, Inc.

Yang, Z., X. He, J. Gao, L. Deng, and A. Smola (2016). Stacked attention networks for image question answering. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21–29.

Yin, W. and H. Schütze (2016, August). Learning word meta-embeddings. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Berlin, Germany, pp. 1351–1360. Association for Computational Linguistics.

Young, P., A. Lai, M. Hodosh, and J. Hockenmaier (2014). From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *Transactions of the Association for Computational Linguistics 2*, 67–78.

Yu, L., Z. Lin, X. Shen, J. Yang, X. Lu, M. Bansal, and T. L. Berg (2018). Mattnet: Modular attention network for referring expression comprehension. In *CVPR*.

Yu, Z., H. Wang, X. Lin, and M. Wang (2015). Learning term embeddings for hypernymy identification. In *Proceedings of the 24th International Conference on Artificial Intelligence*, IJCAI'15, pp. 1390–1397. AAAI Press.

Zaremba, W. and I. Sutskever (2014). Learning to execute. *CoRR abs/1410.4615*.

Zellers, R., Y. Bisk, A. Farhadi, and Y. Choi (2019, June). From recognition to cognition: Visual commonsense reasoning. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Zhao, Z., T. Liu, S. Li, B. Li, and X. Du (2017, September). Ngram2vec: Learning improved word representations from ngram co-occurrence statistics. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, Copenhagen, Denmark, pp. 244–253. Association for Computational Linguistics.

Zhu, Y., M. Elhoseiny, B. Liu, X. Peng, and A. Elgammal (2018). A generative adversarial approach for zero-shot learning from noisy texts. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1004–1013.

APPENDIX A

Hyperparameters

Hyperparameters used in this work are listed below for each experiment. For the tests where hyperparameter optmization is performed, the best performing parameters are underlined.

A.1 Experiments in Table 4.6:

DistilBert:

$$LR = 1e^{-5}$$
$$Epochs = [1, \underline{2}, 3, 4, 5]$$
$$Clipping = 1$$
$$Epsilon = 1e^{-8}$$

Bert:

$$LR = 1e^{-5}$$
$$Epochs = [1, 2, 3, 4, \underline{5}]$$
$$Clipping = 1$$
$$Epsilon = 1e^{-8}$$

A.2 Experiments in Table 4.7:

ResNet:

$$LR = [\underline{1e^{-3}}, 1e^{-4}, 1e^{-5}]$$
$$Epochs = [2, \underline{3}, 4, 5, 6]$$
$$Momentum = 0.9$$

Bert:

$$LR = [1e^{-3}, 1e^{-4}, \underline{1e^{-5}}]$$
$$Epochs = [2, \underline{3}, 4, 5, 6]$$
$$Clipping = 1$$
$$Epsilon = 1e^{-8}$$

A.3 Experiments in Table 4.8, 4.9, and 4.10:

$$LR = 1e^{-5}$$
$$Epochs = 3$$
$$Momentum = 0.9$$
$$Clipping = 1$$
$$Epsilon = 1e^{-8}$$
$$|U|^{1} = 512$$

[1]Attentive pooling network size.

VITA

Erhan Sezerer

Academic Experience

2013–2021 | **Research/Teaching Assistant**
Department of Computer Engineering, Izmir Institute of Technology

Education

2012–2015 | **MSc in Computer Engineering**, Izmir Institute of Technology, Turkey.
Title: News Story Analysis With Credibility Assessment By Opinion Mining
Advisor: Assoc. Prof. Dr. Selma Tekir

2007–2012 | **BSc in Computer Engineering**, Izmir Institute of Technology, Turkey.

Publications

2021 | Erhan Sezerer and Selma Tekir. A survey on neural word embeddings. *arXiv preprint arXiv:2110.01804*, 2021

2021 | Ipek Baris Schlicht, Erhan Sezerer, Selma Tekir, Oul Han, and Zeyd Boukhers. Leveraging Commonsense Knowledge on Classifying False News and Determining Checkworthiness of Claims. *arXiv e-prints*, page arXiv:2108.03731, August 2021

2021 | Erhan Sezerer and Selma Tekir. Incorporating concreteness in multi-modal language models with curriculum learning. *Applied Sciences*, 11(17):8241, 2021

2020 | Elgun Jabrayilzade, Algın Poyraz Arslan, Hasan Para, Ozan Polatbilek, Erhan Sezerer, and Selma Tekir. A turkish topic modeling dataset for multi-label classification of movie genre. In *2020 28th Signal Processing and Communications Applications Conference (SIU)*, pages 1–5. IEEE, 2020

2019 | Erhan Sezerer, Ozan Polatbilek, and Selma Tekir. Gender prediction from tweets: Improving neural representations with hand-crafted features. *arXiv preprint arXiv:1908.09919*, 2019

2019 | Erhan Sezerer, Ozan Polatbilek, and Selma Tekir. A turkish dataset for gender identification of twitter users. 2019

2019 | Erhan Sezerer, Ozan Polatbilek, and Selma Tekir. Gender prediction from turkish tweets with neural networks. In *2019 27th Signal Processing and Communications Applications Conference (SIU)*, pages 1–4. IEEE, 2019

2019 | Erhan Sezerer, Ozan Polatbilek, and Selma Tekir. Türkçe tweetler üzerinden yapay sinir ağları ile cinsiyet tahminlemesi. In *27th Signal Processing and Communications Applications Conference, SIU 2019*. Institute of Electrical and Electronics Engineers Inc., 2019

2018 Erhan Sezerer, Ozan Polatbilek, Özge Sevgili, and Selma Tekir. Gender prediction from tweets with convolutional neural networks: Notebook for pan at clef 2018. In *19th Working Notes of CLEF Conference and Labs of the Evaluation Forum, CLEF 2018*. CEUR Workshop Proceedings, 2018

2017 Erhan Sezerer and Selma Tekir. A relativistic opinion mining approach to detect factual or opinionated news sources. In *International Conference on Big Data Analytics and Knowledge Discovery*, pages 303–312. Springer, 2017

2015 Erhan Sezerer. News story analysis with credibility assessment by opinion mining. Master's thesis, Izmir Institute of Technology, 2015