Machine Ontologies: Approaches to Semantic Modeling in Machine Learning

Morgan J. Weaver
Dept. of Computer Science, Seattle University
Seattle, WA
weaverm1@seattleu.edu | morganjweaver@gmail.com

***Abstract*— As enabling technologies continue to evolve, Natural Language Processing has become increasingly sophisticated, though it is still far from the experience of a natural dialogue. Users interact with Natural Language Processing (NLP)-enabled technologies daily, from Google's predictive search to Apple's Siri. The way in which meaning, function, hierarchy and context are mapped to words and the more complex structures consisting of them will determine the accuracy and success of various NLP tasks, and the way algorithms understand natural language.**

It is therefore crucial to develop high-quality machine ontologies with which to create useful mappings between words and their properties, thus unlocking more sophisticated and efficient embeddings for researchers, industry, and eventually end users. This literature review will comprise a survey of models and semantic architectures used in NLP tasks in machine learning, including their history, mechanisms, use cases, and caveats.

Keywords: semantics; machine learning; deep learning; word embeddings; NLP

I. Introduction

"In the world of the Postmodern, it's not about what you do, but why you do it."—M. Franklin

As of 2018, anyone who has attempted to initiate a nuanced dialogue with one of the popular voice-activated assistants such as Cortana, Google Home, or Siri, has probably noticed that these entities are limited in their understanding of natural language, while science fiction optimistically demonstrates what a natural dialogue with AI would actually sound like, as it has for over 70 years. Why has research failed to produce this level of sophistication? Language in humans is believed to be somewhere between 50,000 and 500,000 years old, with the complexity of language necessary to communicate nontrivial ideas being between 50,000 and 150,000 years old [1]. Language is a complex phenomenon enabled by at least tens of thousands of years of evolution and processed in an organ that is not entirely understood. Naïvely considering the static map of world language lexica and their grammatical rules adequate for creating a working model for machines to fluently interface in natural languages would be like trying to infer the complex interactions of a patch of Amazonian jungle based on aerial photos and a list of plant and animal names. In this way, the *what* of language has historically been more obtainable than the deeper contexts and subtle semantic features of the *why.* Processing must occur in order to extract the latter aspect of language, and is inextricable from language as visual symbol and pattern when language is used for communication. This processing requires a semantic and functional architecture with entities and their relationships within a non-generic domain. In this way, we may refer to these architectures as *machine ontologies* for NLP.

This review will focus primarily on word embeddings as tools for mapping a number of functional-syntactical and semantic-contextual language aspects, as well as the equally important consideration of which use cases, operations, and inferences these embeddings then enable for improved natural language usage and comprehension in machines. Caveats and risks in modeling will follow the literature review. The paper concludes with a high-level summary of the state of word embeddings and semantic architectures in NLP and machine learning.

II. Data Structures

One of the fundamental tools of computational natural language processing, data structures profoundly influence the available operations that can be made on a word or symbol. The term "word embedding" is in practice synonymous with numerical vectors consisting of real numbers. These vectors do not all contain the same representation of information, however.

A simple example of a classic sparse feature representational embedding the one-hot vector. It consists of a vector in which each position corresponds to a single word, with a binary representation of 1 or 0. Only one position is typically set to 1. Words are then represented in a specific and concrete manner. The major problem with this approach being that *each* word in the entire vocabulary must have its own position mapping. So a textbook with 400,000 words would produce one-hot mappings with 399,999 positions set to 0 and a single position set to 1, hence the term *sparse.* This design has the advantage of not weighting a variable with ordinal data, and providing expressive categorical representation, as many machine learning models are not

capable of working directly with categories. This approach is limited and specific in its application, as mappings contain no information about a given word's meaning or context, and is often used as an intermediate step in a multi-part NLP process.

High-dimensional sparse vectors are also used in the *Bag of Contexts* model, where each word is represented by a vector of this type, and can be directly mapped to the context in which a word appears. This representation is used in distributional models, which posit that meaning lies in the statistical frequency of words with other words.

A more frequently-used approach to representation with vectors consists of using denser real-valued vectors, in which the series of values represent latent continuous features in a high-dimensional space. This representation is in practice the most synonymous with the term *word embedding.*

III. Embedding Model Categories

A. Frequency-Based

Also referred to as *count-based,* these models are unsupervised, meaning that no labeled training data is required to train the models. Frequency-based refers to the compilation of statistics about different words in a training corpus, and their relationships and probabilities regarding other words, especially neighboring words. These relationships are often summarized in a co-occurrence matrix, which is never used directly, but produces useful embeddings once factorized with other data. These models usually produce small, dense vectors that treat words found in similar contexts as semantically similar. The accuracy of these embeddings is often dependent on a very large corpus of text, relative to predictive models.

B. Predictive

Predictive models are less concerned with the description or mapping of words, as their goal is to *predict* the probability of a word and its neighbors, with frequencies of words being treated as a means to an end. The model incrementally adjusts its weights and, by proxy, the resulting word vectors to maximize the chosen goal. These goals depend on the implementation, and, to provide two common examples, could be prediction of a word given its context or neighboring words, or prediction of a word's context (either semantic or morpho-syntactic) given a variable-sized window of neighbors. As mentioned, these goals are only instruments with which to influence the semantic architectures, or ontologies, of the resultant word vectors, whose latent-space representations (again, consisting of semantics and morpho-syntactic qualities) will vary depending on the rationale with which they were created. Unlike frequency-based models, predictive models require labeled training data, which can be costly in terms of human effort and time.

It is important to keep in mind that excellent implementations of both frequency-based and count-based models exist, providing high accuracy and precision in various tasks. As in other areas of data science and statistical modeling, choosing the "best" model is less about popularity, and more about choosing the appropriate tool for the job—*id est.*, the data constraints and qualities, and the objective of the researcher's endeavor.

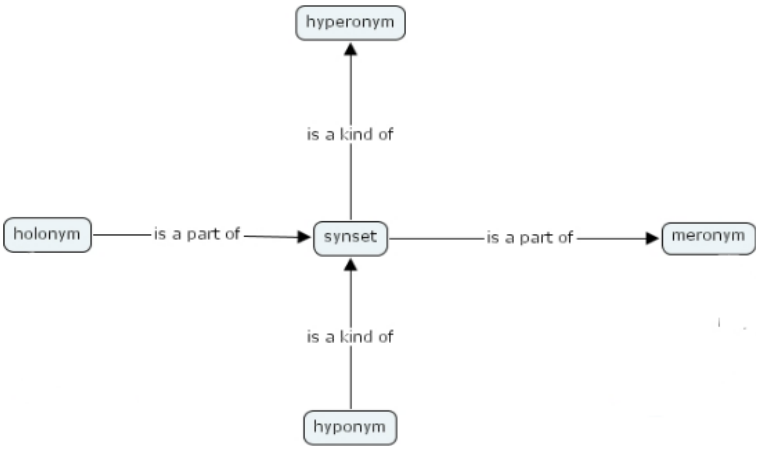

Figure 1: Semantic concept diagram depicting the relationships between a given word and its synset.

IV. Selected Terminology

There are a few terms used in the literature in describing the semantic (Figure 1) and linguistic qualities of words that are somewhat specific to their field. The most relevant of these are:

- *Holonym*: member of a word's *morpheme family* that captures a broad-to-specific "is a part of" relationship. For instance, *chinchilla* a holonym of *fur* and *tail*. *London* is a holonym of *Great Britain.*
- *Meronym*: describes a word's "is a part of" relationship in the opposite direction—*whiskers* and *ears* being meronyms of *chinchilla*. *Lisbon* being a meronym of *Portugal.*
- *Hypernym*: describes an overarching concept or category under which other words are subsumed. For instance, *Rodentia* is a hypernym of *chinchilla*.
- *Hyponym*: term denoting a word's relationship to the set of words more categorically granular words it encompasses. For instance, *chinchilla*, *gerbil,* and *porcupine* are all hyponyms of *Rodentia*.
- *Morpheme*: Describes a morphological (as opposed to semantic) unit of language that cannot be further divided, such as *xeno-*, *morph*, and *–ic*. Some morphemes can be used alone, while others must be combined with other morphemes to disambiguate their context, like the suffix *–s* to imply plurality of a noun.

V. Literature Survey

The following is a broad and incomplete survey of current literature that meets the criteria of relating to the topic of semantic paradigms for natural language processing tasks in machine learning.

The papers herein were curated with the objective of providing breadth, rather than depth, to the reader. Both state-of-the-art paradigms and foundational paradigms are referenced to provide a more intuitive understanding of this topic. For a focused overview of recent trends, see references [16].

A. Syntactic Dependency Parsing

Seminal work by Mikolov et al. (2013) [10] explored negative sampling to enhance arbitrary contexts in word embeddings, in which only a small number of weights in the hidden layer of a neural network are updated with each sample as one-hot vectors are passed in, as n-1 items on a one-hot vector are set to zero to begin with. This technique reduces the amount of time and compute needed to train the models, and produced in higher-quality vectors, which in turn resulted in greater test accuracy.

Levy and Goldberg (2013) [7] take this paradigm a step further by incorporating syntactic context using automatically-produced dependency parse-trees to enhance the existing skip-gram model. Skip-gram models are known to be useful in inferring broad topical information, rather than syntax. The researchers created a skip-gram model which was switched from classic window sampling to a head-target-modifiers sampling method, focusing on the subject of the sentence and its modifiers within the window of the whole sentence rather than a preset window. The model is supported by the ability to automatically label various parts of speech such as modifiers, nouns, subject versus object, etc., and collapse the sentence into its most relevant details. The resulting customized or dependent windows are maximized for relevance and enhanced with syntactical information.

Comparison of this model with existing Bag-of-Words models with windows of both 2 and 5 showed that BOW returned meronyms, or words that were associated with a target word as parts of a whole, rather than cohyponyms, or words that behaved like, or were semantically equivalent to the target word, as Dependency-based embeddings do. For instance, given a word like UK, standard BOW might return London, Scotland, and Westminster--domain-related meronyms--whereas Dependency-based would return other countries, which behave in a syntactically equivalent manner, such as France, Japan, and Uganda.

As a byproduct of the dependency trees, the researchers were able to pierce the opaque nature of neural word embeddings by exploring which contexts most strongly "activated" a given word. For example, the top three activators of the word "Turing" were "machine", "test", and "theorem".

B. Semantic Hierarchy Generation

In *Learning Semantic Hierarchies via Word Embeddings*, Fu et al. of Baidu [5], one of the world's largest online retailers, detail their model for automating semantic hierarchy construction using word embeddings. The researchers claim that their model outperforms the current models (as of 2014), and provide experimental results for their model, which is compared to the hierarchies assigned by two human raters from a dataset of 418 entities.

The researchers note that semantic hierarchies are the main components of ontologies, and that the ability to create these hierarchies is synonymous with the ability to organize large bodies of knowledge to countless ends. This process is limited in scope by painstaking manual construction, and by our limited abilities to consume and remember information, and provides a fitting opportunity for automation with NLP and machine learning. As vector-based word embeddings (such as those created by Skip-gram and CBOW models) preserve analogical information, they have also been found to preserve hierarchical information. The researchers verify this assertion by creating their own hierarchical vector offsets from training data, and show that the resulting word relations are clustered. This is an indication that the found hypernym-hyponym relations are appropriately related by their vector offsets in latent space, which can be loosely conceptualized as having clusters of spatially-mapped meaning--for instance, one would expect *sloth* and *vertebrate* to be closer in latent space than *sloth* and *planet*.

Two main semantic extraction methods are proposed based on projection matrices, the latter being a refinement of the first, and the ultimate model the researchers chose to implement[1]. The semantic extraction method consists of a piecewise linear projection in which training data is first clustered into groups in latent space rather than taken as a whole. The researchers then work from the principle that for every word and its hypernym in the training corpus, there exists a transition matrix Φ that maps a given word to its correct hypernym, within the spatial limits of its predefined semantic cluster in latent space. Using a modified version of Φ , separate projections are learned within each word cluster. To create initial projection matrices, labeled training data is obtained from the Chinese semantic thesaurus Tongyi Cilin, which offers five levels of semantic hierarchy and contains over 100,000 words. Hypernyms are determined by selecting the hypernym candidate with a Φ[2] that places its embedding vector closest to the specified radius of a chosen hyponym in vector space. This provides a supervised model with which to train the initial model.

For the test condition, the researchers implement a Skip-gram-based model to provide the initial embeddings from which to initialize the model, based on a 30-million sentence corpus provided by Baidu. The trained model's projection

---

[1] The researchers found that a obtaining a consistent Φ was difficult, and offer their own modified version, which minimizes the mean squared error and utilizes multi-dimensional vectors rather than scalar values.

[2] Many studies mention using cosine similarity to measure a word's correlation to another word in latent space. Cosine similarity captures the angle between the two words rather than the absolute distance in vector space and creates a more useful metric than a magnitude or Euclidean distance-based metric. Note that Φ measures the correlation between binary values rather than vectors, and can be thought of as similar to the Pearson coefficient.

learning data is derived from the previously mentioned thesaurus Tongyi Cilin. Two human raters create hypernym-hyponym pairs for 418 entities selected at random from the Baidu test corpus against which to test the model's pair results.

A completely manually built hierarchy incorporating both Tongyi Cilin's classifications and pre-existing category taxonomy of Wikipedia achieves an F-score of 73%, which determines accuracy by combining both recall and precision into one metric for binary classification tasks. Five other common NLP models are tested and similarity scores extracted. F-scores ranged from 35%-62%. The proposed method based on word embeddings, trained on the Baidu corpus, achieves an F-score of 73.74%. This embedding-trained model is then combined with Tongyi Cilin manual hierarchy, and all positive results from the two methods are merged and used to infer transitive relationships between various pairs--this results in an F-score of 76.29%. Finally, The word embedding-based model is combined with manually-created Tongyi Cilin and Wikipedia taxonomy data to provide an F-score of 80.29%. This is a significant improvement relative to the state of the art hierarchy extraction model, which scores 62.13%.

Fu et al. provide a highly performant new model with which to create semantically accurate and precise hierarchies. From categorizing thousands of consumer products to organizing encyclopedic information, there are many potential uses for such a model.

C. Relational Mappings Continued: Sparse vs. Explicit

Building on relational similarities arising from vector arithmetic [2], Levy and Goldberg (2014) [8] expand on this concept by devising two new methods of extracting relational similarities. The researchers first develop a semantic relational extraction method, coined as PairDirection, which takes vector direction and ignores Euclidean distance. The researchers then show that solving analogy tasks with vector algebra is equivalent to maximizing the linear combination of three pairwise word similarities in analogy tasks, and apply a logarithmic normalization of the similarity score calculation used in previous research. While the default model is additive, the second of the two new models is multiplicative, and amplifies the delta between minor differences in score while minimizing larger differences. This method ultimately yields more semantically sensitive performance in analogy tasks by reducing dominant properties of words and amplifying lesser aspects, so that words that are disproportionately weighted with one aspect are less likely to fall outside of their expected analogy sets.

The second contribution of this paper consists of a demonstration of analogy tasks using distributional (or sparse-vector) representations as opposed to predictive models, which produce dense-vectored neural word embeddings mapped to latent space. This is a significant achievement, as the bag-of-contexts model was previously thought to be less capable of preserving relational similarities than implicit, or neural embeddings.

The authors test their multiplicative relational extraction method against the existing additive method, on both neural embeddings and explicit embeddings. Performance between the two embedding types was shown to be equivalent on the SemEval dataset, consisting of 79 semantic relations, for both additive and PairDirection models. PairDirection is a model developed by the researchers that takes direction rather than distance into account. While this model did well with restricted-vocabulary semantic tasks, it achieved less than 1% accuracy on the MSR and Google datasets (both used by Mikolov et al. in previous research), which included morpho-syntactic analogy questions either exclusively or in addition to semantic analogy questions.

The multiplicative relational extraction was favored above the directional, and outperformed the additive model overwhelmingly. Between explicit and embedding representations, the multiplicative method performed better on one or the other depending on the analogy type, with an almost perfect 50/50 split between analogy types. For instance, embedding representations were found to be better in currency, family, verb, and past tense analogies, while explicit data was found to be better for country (with 99.41% accuracy), capital city, adjective and noun analogy tasks. One interesting advantage of explicit, sparse vectors is their ability to let us extract the top features of different aspects by pointwise multiplication of two word representations sharing a particular aspect (thus looking at their vector intersection in high-dimensional space). What does this look like? The researchers demonstrate aspects such as Female, Royalty, and Currency. For Currency, Yen and Ruble were intersected to show the top features of devalue, banknote, denominated, billion, banknote, pegged and coin.

This close examination of relational extraction with attention to vector direction and normalization, in addition to a demonstration on both explicit and implicit representational models broadens the previous research meaningfully, offering newer and higher-performing relational extraction and inspection paradigms.

D. Combination Models

In their ground-breaking paper, Pennington et al. (2014) [12] introduced the NLP and deep learning communities to GloVe, short for Global Vectors, a global log bilinear factorization method that combines n-gram window (as seen in skip-gram, etc.) and global matrix factorization methods (as seen in latent space-based representational models) to create word embeddings featuring the advantages of both models. Much as Mikolov and colleagues throw away a large portion of zero-scores from their sparse vectors during model training in negative sampling, Pennington and colleagues remove zero-elements from their word-word co-occurrence matrices to create a faster, more efficient, and performant model relative to existing models as of 2014.

The model presented in GloVe combines the utility of two major embedding two models. The first is an unsupervised global corpus frequency summarization, which results in a large word-word co-occurrence matrix revealing statistical information about the corpus, especially in terms of word ranking. The matrix produced is processed to reveal

not just word frequencies but also co-occurrence probabilities, and is especially sensitive to ratios of co-occurrences when comparing the ratios of different related words, while unrelated words tend to have a co-occurrence value closer to 1. The matrix is then used to train a specific weighted least-squares regression model, with vectors based on ratios rather than initial probabilities.

GloVe's performance was then tested on the canonical Mikolov et al. (2013) analogy task set divides into semantic and syntactic subsets, as well as five different word similarity task sets, and finally the CoNLL-2003 English benchmark test for named entity recognition (NER), which includes persons, locations, organizations, and miscellaneous named entities in an array of Reuters news documents. As in Levy and Goldberg (2014), cosine similarity is utilized to produce similarity scores for the words in the similarity tasks, and negative sampling, as mentioned in foundational work by Mikolov et al. GloVe performed competitively next to a variety of singular value-based vector models (SVD), as well as CBOW and Skip-gram models. In the analogy tasks, GloVe consistently outperformed all other models in semantic tasks, and outperformed other models in syntactic tasks in half the training sets by corpus size, excelling at the low end (a 1.6 billion word corpus) and the high end (a 42 billion word set). In word similarity tasks, GloVe was compared to the same models in a 6 billion word corpus and to a log-adjusted SVD model (SVD-L). GloVe outperformed all models for every similarity task at the 6-billion corpus size with the exception of Stanford's Contextual Word Similarities (SCWS), and outperformed SVD-L on the 42 billion word corpus. GloVe is then compared to 8 other models including the aforementioned, plus several newer matrix factorization models. GloVe outperformed all other models on 3 out of 4 task sets. Finally, GloVe is compared to the then-state-of-the-art log bilinear model, word2vec, and outperforms word2vec by a significant margin for both skip-gram and CBOW versions of word2vec on word analogy tasks.

Across the literature, GloVe is considered to be one of the most cutting-edge unsupervised word embedding models available. However, as of the end of 2017, there are still no embedding training models that have superseded word2vec's fundamental design, published in 2013, although plenty of word2vec variants, such as GloVe, continue to make incremental advances over the original word2vec.

E. Morphemic Compositionality and Semantics

The same year Mikolov et al. published *Distributed Representations of Words and Phrases and their Compositionality* (heralding Word2Vec), the Stanford NLP department [9] published its own seminal work on morphemic semantics and the compositional properties therein. Prior to 2013, no work of this exact nature had been released, though the authors reference a 2003 study, which tended to cluster words with the same suffix by class. In contrast, this paper was published after 2006, in a post-Hinton et al. (2006)[6] world, in which training neural networks in reasonable amounts of time and compute became feasible with the use of Restricted Boltzmann Machines.

Luong et al. introduce the community to a new machine ontology of morphemics for NLP that captures the semantic impact of morphological features in a useful and extensible way. The model has the advantage of being able to handle unseen and rare words by creating latent space-mapped morpheme families from which to compose more complex meaning than models that treated words as whole entities. This is an abrupt departure from character-level embeddings, which treat text generation and character-level probabilities as statistical-descriptive paradigms that may capture morphology and some syntax, but not semantics--characters are nothing but symbol patterns without semantic value. A quick glance at the amusingly RNN-generated MealMaster recipes will remind the interested researcher that context is not captured when a recipe instructs us to "Sauté the peas in the refrigerator for at least 8 hours"[16]. Though frequent patterns are caught, semantics are not. While latent space can capture some syntactical relationships, such as plural vs. singular with the help of vector algebra, these relationships are far more difficult to capture for rare and complex words for which there is little prior information.

In what is termed the morphoRNN model, words are broken down into components, with each word being treated as a recursive stem-affix composition until the words can no longer be decomposed. The affixes are treated as intercept vectors in high-dimensional space for the word stems, or non-affixes, which may or may not be completely decomposed. The parameters for the RNN to learn become the intercepts, the broader morphemic embedding matrix, and the morphemic parameters composed of stems and affixes. This model builds upon the previous word-level RNNs.

The authors first explore an implementation devoid of contextual information, with input consisting of a reference embedding matrix of word vectors gleaned from Wikipedia, of which several versions were used for comparison. The authors then create a context-sensitive model which incorporates NLM learning as a layer on top of the RNN, which in turn backpropogates adjustments all the way down to the morphemic layer. Both models complete a forward pass, recursively constructing a morpheme tree at the word level, followed by the aforementioned backpropagation pass which discerns the gradient of the utility function with respect to the given parameters as mentioned above. The NLP software Morphessor is used to determine where splits should occur, and to label the subparts as prefixes, stems and suffixes as appropriate. The words are post-processed to split the compound stems further where possible, split hyphenated words, and determine what to do with other edge cases. Two sets of existing Wikipedia word vectors, one incorporating multiple vectors for individual words, were used in testing. For the context-sensitive model, a 10-word window was used.

Two experimental tasks were chosen to test the context-sensitive and contest-insensitive models. One involved word similarity scoring of a broad range of common words, consisting of several pre-existing word similarity task

datasets which include scored assigned by 10-51 human raters each. The second set involved rare words as determined by frequency range in the 2010 Wikipedia corpus, with scores obtained from human raters via Amazon Mechanical Turk. In the second data set, a rare word was chosen, and an array of its relations--namely meronyms, hyponyms, homonyms, and attributes--were collected. A second word was randomly chosen from this collection, and the process was repeated to generate a total of two pairs for each rare word. In the first task, Spearman rank correlation with human scores ranged from 32.97%-71.72% depending on dataset and model, with the context sensitive model outperforming the context-insensitive model, though only narrowly in some cases, every time. In the rare word task, Spearman correlations with the Mechanical Turk-obtained ratings ranged from a low of 14.85% for context-insensitive to 34.36% for context-sensitive. The context-sensitive model, unsurprisingly, tended to balance syntactic and semantic characteristics more equitably than the context-insensitive model.

The RNN and NLM combination handles morphemic compositionality modeling in a novel manner, tackling a task that had not been well-addressed as of 2013. With an RNN handling the mapping of morpheme vectors in latent space, and the NLM utilizing the contexts of neighboring words in the training corpus, Luong et al. contribute a thoughtful new model, and a starting point for subsequent research in morphemic compositionality of words and its semantic implications in machine learning and natural language processing.

F. Compositionality of Semantic Character Sets

Building upon Luong et al. (2013), Chen et al. in the 2015 paper Joint Learning of Word and Character Embeddings [4] offer a compositionality-sensitive model designed for the semantically rich set of characters in the Chinese alphabet. As each character is a full word, with most words being combinations of two or more characters that are in turn semantically relevant to the word they compose, character-level learning becomes far more useful than in the Roman alphabet, where most characters do not have individual meaning as words.

The authors choose a CBOW-based model termed Character-Enhanced Word Embedding (CWE) to embed their training corpus, where the goal of training is to predict a word given several context words (skip-gram, as a quick refresher, accomplishes the inversion of this task). The complexity of the objective is increased due to the contextual role of many Chinese characters, which can be ambiguous and depend on their neighbors for their own semantic meaning, just as many English words can have several senses that require context to disambiguate (got, class, and stream are three examples of words with more than one definition). The authors also note that a small percentile of Chinese characters are transliterated, referring to non-Chinese words, which represent phonetic information rather than semantic. With these constraints in mind, CWE preprocesses compound character-based words to arrive at semantically-enhanced word-level representations that are enriched by the vectors of their constituent characters.

To this end, the authors create multiple vectors for each character and explore position-based, cluster-based, and nonparametric methods for choosing among vectors when assigning contextual significance. Testing these models consisted of both word relatedness tasks and analogy tasks, with CWE being compared to baseline models including vanilla CBOW, skip-gram and GloVe. The authors note that this model can easily be incorporated into various neural network models and matrix factorization models, including those that it is compared to in the experimental tasks.

The authors consider three different means of addressing the ambiguity of many Chinese characters, to which they refer to as Multiple-Prototype Character Embedding. The first of these is Position-based, which chooses a vector based on the position of a given character in a word to remove the ambiguity of the contextual information encoded in its order. The second relies on a more cumulative approach, in which *all* instances of a character are clustered according to its context to form several prototypes for that character, and its most common embedding is incorporated in its new embedding. Though this is not the same as Position-based embeddings, the authors note that it can be combined with these embeddings. The final model for addressing word ambiguity is termed Nonparametric Cluster-based embedding, where the clustering concept is modified to accept a dynamic and incrementally increasing number of clusters during training, rather than completing a post-hoc analysis of the clusters as in k-means clustering.

In their analysis of these models, Chen at al. choose two classic tasks: word similarity and analogical reasoning, such as the oft-cited *king - man + woman = ?* in which the answer should be queen. The corpus chosen consisted of news articles from *The People's Daily*, with 31 million words, 105,000 unique words, and 6,000 unique characters, notably covering 96% of the characters in the national standard charset as defined by the Chinese government. Vector size was set to 200, with a 5-character context window. As in related work, negative sampling was applied, set to 10 words, in addition to hierarchical softmax. The authors separate their model into plain CWE, CWE with position, CWE with clustering, CWE with position and clustering, and nonparametric CWE. CBOW, Skip-gram and GloVe were used as baselines for comparison, set to default parameters but using the same vector dimension.

In the similarity task, approximately 550 word pairs from two different word similarity test sets were used, all of which had human rankings for comparison. In the Spearman correlation with human rankings, CWE with clustering outperformed all other models by margins of approximately 5%-20%. In one case, nonparametric CWE outperformed CWE with clustering by about a tenth of a percentile. For all cases, all CWE models were usually within 1-3% of one another, and uniformly in the high 50th percentile to low 60th percentile in terms of correlation with human scores.

In the analogy task, performance was not compared to human raters, as every analogy instance included a correct answer. CWE and CWE with position models were

incorporated into CBOW, Skip-gram, and GloVe, and the non-CWE version of each model was separately tested. Analogies were chosen for their lack of ambiguity, and categories consisted of countries' capital cities, states/provinces associated with cities, and family roles. Of the four categories, Skip-gram CWE with position outperformed regular Skip-gram by just under 2 percentile points in the state/province category with 84% accuracy, and outperformed regular Skip-gram by just under 4% for family roles. In the capital cities of countries task, GloVe CWE with position outperformed regular GloVe by about 3%, and was also the model that performed best overall for the three tasks, narrowly surpassing Skip-gram CWE with position by under 1%.

The authors remind non-Chinese speakers that Chinese text relies heavily on sub-word composition for its semantic content, so building upon the limited amount of previous work on the semantic implications of morphemic compositionality is critical for developing models appropriate for parsing Chinese text specifically. The CWE model, which can be incorporated into existing models, offers an enhanced version of semantic embedding to improve the quality of word embeddings, with the added benefit of relieving words in context of some of their ambiguity. One of the more interesting avenues for future research added by the authors consisted of assigning weights to the characters within words for added semantic sensitivity.

VI. Algorithmic Bias

Any good model is a synergy of theory and praxis, and word embeddings are no exception. While algorithms themselves are not necessarily biased, the data selected to train them can create inaccurate, imprecise, and ethically problematic models when applied to social tasks.

As models become more powerful, they have in many cases become capable of processing more data than their users can review. Many models require very large corpora of data for testing and training in order to create realistic embeddings. One of the hidden pitfalls of this scenario is the potential for bias in the data sets, which leads to bias in the resultant embedded vocabularies.

One excellent example of this was described in the aptly titled *Man is to Computer Programmer as Woman is to Homemaker? Debiasing Word Embeddings*, by Bolukbasi et al. [2]. The researchers demonstrated that word embeddings created with a seemingly neutral training data source--Google News articles--were remarkably sexist in the context of gender-neutral words. One example consists of the uneven distance in latent space between words like *nurse* and *cupcake* to female/woman-associated words like *girl* or *matron*, whereas other gender-neutral words like *chef* and *pizza* were more proximal to male-associated words, even though the morphology of the neutral words does not explicitly encode gender. This is what is referred to as *direct bias*. Indirect bias is the second-level association of supposedly gender-neutral words with other gender-neutral words, like *teacher* being closer to *softball* than *wrestling*, owing to the feminine bias inherent in the word *softball*. The researchers note the obvious utility of preserving gender-dependent words, while readjusting the most biased gender-neutral words to be less representative of the sexism expressed in the training data.

One notable example is demonstrated in the semantic associations of names used in advertising. Dr. Latanya Sweeney, Professor of Government and Technology in Residence and Director of the Data Privacy Lab at Harvard, was famously being interviewed by a journalist, when a Google search result for her name returned several sponsored ads for arrest record search services. Dr. Sweeney, with no criminal record herself, conducted a study [14] which found that names that are nearly exclusively given to African-Americans were up to 25% more likely to return criminal record-related ads, with misleading titles such as “Criminal records, phone, address, & more on Latanya Sweeney.”

This bias has the potential to negatively affect anyone whose name fits such a criteria, whether being considered for a scholarship, award, or job, or anyone who holds a position of trust, such as a doctor, priest, or professor.

Another excellent example of algorithmic bias was provided in Buolamwini and Gebru [17], in which the authors test three commercial facial gender classifiers by Microsoft, IBM and Face++. Accuracy was found to be highest for white, male subjects, and lowest for dark-skinned, female subjects in Microsoft and IBM's classifiers. Error rates between best and worst-classified groups were found to be as high as 34.4%.

Facial recognition software is used as a security and identification feature in consumer electronics and healthcare. It is also used in law enforcement to identify potential criminals, with disturbing implications—there is already at least one incidence of an innocent person being sentenced to jail due to the inherent bias in the models[3]. The facial recognition software specific to law-enforcement is offered by companies CyberExtruder and Vigilant Solutions, who as of December 2017 (4 years after Sweeney’s study was released) had not tested their software for bias upon being contacted by the journalist reporting on the issue [3]. Though not explicitly a semantic model in the realm of NLP, this examples were chosen to clearly illustrate some of the real-world implications of algorithmic bias.

These implications are not only offensive to broad swaths of society on a global scale, but are also potentially dangerous to *everyone,* with a disproportionate impact on historically disempowered, disenfranchised, and otherwise underprivileged demographic groups. Whether the consumer is an 8-year old girl of high socioeconomic status having a conversation about her future career options with a digital assistant, or a person of color wrongfully implicated in a criminal investigation as cited above, algorithmic bias is a far-reaching problem with both explicit and insidious capacity to cause harm.

Fortunately, researchers can avoid this issue quite simply--by screening initial data, probing trained models for different types of bias, and finally by making an effort to create diverse research teams with multifaceted perspectives and experiences. These simple actions can reduce the

potential for harm while creating higher-quality research and enabling more rational decision-making[13].

VII. Conclusion

Myriad options for semantic modeling are currently available at the intersections of Machine Learning and Natural Language Processing. This research is relatively new, and has only recently been further enabled by novel methods allowing efficient training of neural networks for the production of high-quality embeddings, as well as the availability of very large corpora of training and testing data, as well as quality labeled datasets for use in supervised training.

In this literature review, we cover several models for creating word embeddings and consider the semantic and morpho-syntactic or functional mapping structures they enable. We cover advances in syntax-based semantic dependency parsing, in semantic hierarchy generation, in a hybrid embedding paradigm and its resulting embeddings, in morphemic composition and its semantic implications, in joint character- and word-level parsing in an alphabet with semantically rich characters. Finally, we consider matters of data hygiene and the potential for algorithmic bias, and how this can be avoided. While there are numerous exciting and innovative models to explore that fall outside the scope of this paper, the author has attempted to highlight a variety of interesting, contemporary, and useful models for reasoning about the possibilities in this research area.

Acknowledgment

The author would like to express her thanks to mentor Leo Dirac for guidance on contemporary embedding models and data structures.

References

[1] Bolles, E.B. (2009). How old is language? [Blog post]. Retrieved from http://www.babelsdawn.com/babels_dawn/2009/10/how-old-is-language.html.

[2] Bolukbasi, T., Chang, K.-W., Zou, J., Saligrama, V., and Kalai, A. (2016). Man is to computer programmer as woman is to homemaker? Debiasing word embeddings. *Advanced Neural Information Processing Systems*. (2016), 4349–4357.

[3] Breland, A. (2017). How white engineers built racist code – and why it's dangerous for black people. [Blog post] Retrieved from https://www.theguardian.com/technology/2017/dec/04/racist-facial-recognition-white-coders-black-people-police

[4] Chen, X., Xu, L., Liu, Z., Sun M., and Luan, H. (2015). Joint learning of character and word embeddings. In *Proceedings of the International Joint Conference on Artificial Intelligence*(IJCAI 2015), 1236-42.

[5] Fu, R. et al. (2014). Learning semantic hierarchies via word embeddings. In *Proceedings of the 52th Annual Meeting of the Association for Computational Linguistics: Long Papers*(1).

[6] Hinton, G.E., Osindero, S., and Teh, Y. (2006). A fast learning algorithm for deep belief nets. In *Neural Computation*18:1527–1554.

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

[8] Levy, O. and Goldberg, Y. (2014). Linguistic regularities in sparse and explicit word representations. In *Proceedings of the Eighteenth Conference on Computational Language Learning*(2014), 171-80.

[9] Luong, M., Socher, R., and Manning, C.D. (2013). Better word representations with recursive neural networks for morphology. In *Proceedings of the Seventeenth Conference on Computational Language Learning* (2013).

[10] Mikolov, T., Sutskever, I., Chen, K., Corrado, G. and Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems* (26), 3111–3119.

[11] Nathan M., and Lee N. (2013). Cultural diversity, innovation and entrepreneurship: Firm-level evidence from London. In *Economic Geography* 89(4), 367–394.

[12] Pennington, J., Socher, R., and Manning, C. D. (2014). Glove: Global vectors for word representation. In *Conference on Empirical Methods in Natural Language Processing*(14), 1532–43.

[13] Rock, D. and Grant, H. (2016). Why diverse teams are smarter. In Harvard Business Review. [Blog post] Retrieved from https://hbr.org/2016/11/why-diverse-teams-are-smarter.

[14] L. Sweeney, (2013). Discrimination in online ad delivery. In *Communications of the ACM* 56(5) 44–54.

[15] Young, T., Hazarika, D., Poria, S., and Cambria, E. (2017). Recent trends in deep learning based natural language processing. arXiv preprint arXiv:1708.02709.

[16] Brewe, T. (2015). Do androids dream of cooking? [GitHub] Retrieved from https://gist.github.com/nylki/1efbaa36635956d35bcc.

[17] Buolamwini, J., and Gebru, T. (2018). Gender shades: Intersectional accuracy disparities in commercial gender classification. In *Proceedings of Machine Learning Research* 81, 1–15.