PhD Research Proposal

Lyndon White

15th September 2015

School of Electrical, Electronic and Computer Engineering
Faculty of Engineering, Computing and Mathematics
The University of Western Australia

A. Project Title and Summary

A.1. Project Title

Semantic Vector Representations of Natural Languages

A.2. Summary

The research will investigate methods for the production and utilization of vector representations of natural language preserving meaning. Algorithms producing vector embeddings of sentences and longer documents currently exist, however the field is still developing. Existing methods have not been shown to sufficiently preserve meaning in the vector representation. There has also only been limited investigation in to reversing the projection and resynthesis text from the embedding space. The aims of this project are thus:

- Develop a method for producing semantically consistent vector representations of sentences. This includes evaluating and extending existing methods, and developing new ones.
- Develop a method for resynthesizing text from such vector embeddings. This may be in the form of an extension of a current methods, if they meet the previous aim, or developing new algorithms with the capacity inherent.
- Utilize algorithms in the vector space, to carry out tasks in the natural language space.

B. Research Project

B.1. Background

Over the last five years, word-embeddings have revolutionized Natural Language Processing (NLP). A word embedding is the conversion of a word, into a vector in semantic space. This has several applications and is used to achieve the state of the art solutions to many NLP problems. More recently sentence embeddings have attracted some attention. Sentence Embeddings have also produced state of the art results in there application area. This project aims to create semantically consistent sentence embeddings suitable for using in Natural Language Understanding and Generation (NLU and NLG).

NLP is a key area for modern development. Vast amount of information exists written, or spoken, in natural languages such as English and Chinese, NLP is concerned with processing this. As the amount of information constantly grows, so to does the need to be able to process it computationally. By embedding sentences into a vector space, spacial methods and intuitions can be applied to this processing problem. NLU is the subfield of NLP concerned with creating software which can (to some extend) comprehend the meaning of natural language input. NLG is the field concerned with using software to produce natural language output. Embedding and resynthesizing sentences into and from the vector spaces can be applied to NLU and NLG problems respectively. This combination adds a new angle of attack upon an array of current problems.

Full cycle vector embeddings of sentences would be able to accomplish many tasks which currently require manual intervention. An example how they can be used for abstractive summarization is shown in Figure B.1. Other tasks which could be performed similarly include: paraphrase generation, machine translation and creating descriptions from images. However, currently only limited progress has been made towards the resynthesis step. With just the embedding step state of the art results have been reached in the correspondent tasks: Extractive summarization[1, 2], paraphrase detection [3], similarity measurement for machine translation purposes[**?**], and identifying images based on description[4]. This project proposes to allow the extension of the later method for the latter tasks, to complete the former described tasks.

B.1.1. Problem Statement

Word Embeddings are reversible – it is possible to convert back from a embedding vector to the most similar word. This reversibility is essential for many applications. Often the reversibility is achieved via a nearest neighbor search via embedding the whole vocabulary, this can not be done for sentence vectors as the 'vocabulary' of sentences is far too large. Thus current methods for phrase embeddings are not so trivially reversible. Arbitrary phrase vectors can not be converted into a natural language sentence. A key requirement for being able to synthesize a sentence from a sentences vector is for those vectors to be semantically consistent in the first place. Recent results have indicated that current methods may not be sufficiently consistent in their mapping from meaning to position to meet these requirements. This proposal is to devises new methods which do, and to use them to for bidirectional conversion between vectors and sentences.

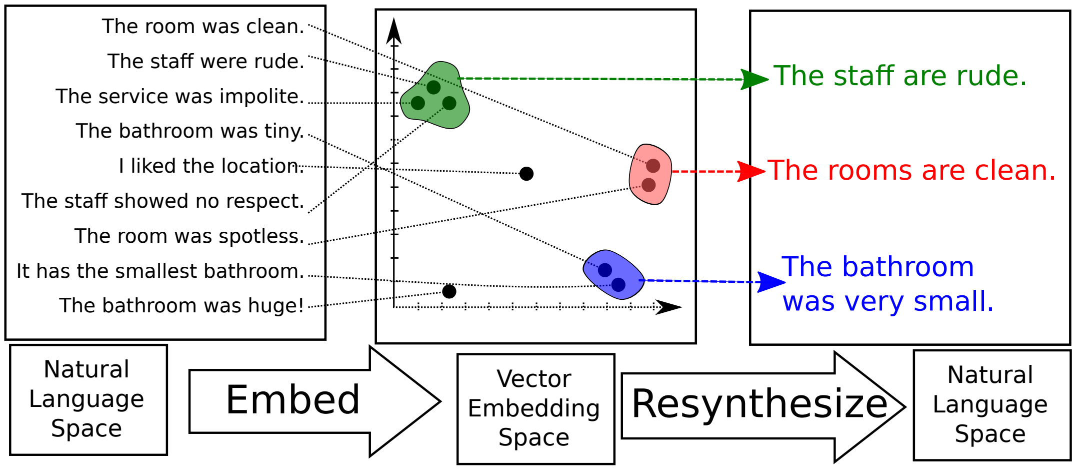

Figure B.1: Workflow for how embeddings may be used to perform abstractive summarization. Sentences (in this case hotel reviews), are taken into the vector space (shown in 2 dimensions for readability, in actuality 50–300 dimentions) where spacial methods are used to cluster commonly occurring meanings, and to disregard outliers. In the Resynthesize step, new sentences are generated which surmise the meaning of each cluster.

Research Question How can the meaning of a sentence be represented as a vector, such that a vector can be resynthesised into a synonymous sentence.

Significance The production of such reversible embeddings is will enhance current NLP techniques to allow for whole phrases to be handled as vectors. Further as other methods for solving word embedding problems – such as short phrase embeddings, and word-sense embeddings – are developed, the extention the reversible phrase embedding methods proposed here will to be obvious and beneficial – as the proposed methods build upon the existing word embedding technologies.

B.2. Literature Review

Figure B.2 shows a rough outline of the development of the array of methods used for generating embeddings for natural langauge processing. In the following sections the methods are broken down by application.

B.2.1. Parsing: RvNN

Recursive Nerual Networks While parsing is a syntactic task, rather than a semantic one this proposal is concerned with, it was the first task to which the Recursive Network (**RvNN**) was applied to. Table 1 shows the performance of the methods. The RvNN

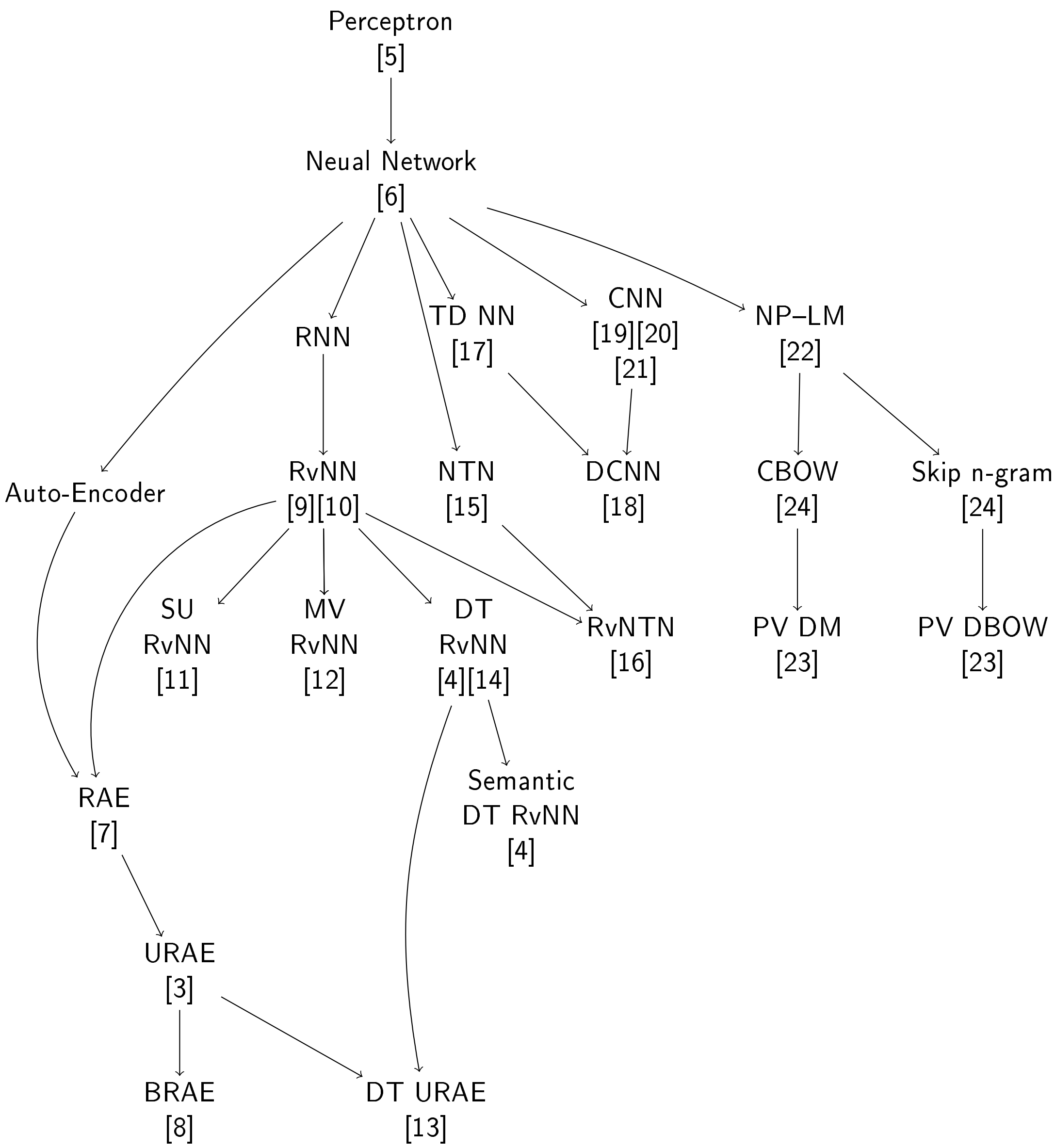

Figure B.2: The “Family Tree” NLP neural networks. The abbreviations are exanded upon in the text below (see **bold** markings)

Year	Author	Method	Performance
2011	Socher et. al.[10]	RvNN	F1: 90.29%
2013	Socher et. al.[11]		Acc: 85.0%
2013	Socher et. al.[11]	SU RvNN	Acc: 90.4%

Table 1: The application of RvNN decended technologies to parsing. In all cases the Test and Training data was the Wall Street Journal Sections of the Penn Treebank.

generalizes the reused of the output as an input, which is present in the Recurrent Neural Network (**RNN**) to be performed over a tree of inputs, with each layer merging into the next. The lowest level input is word vectors, which are stored in a look up table keyed from the word, and trained during network training. They can be initialized randomly or using word embeddings from another model. In the case of the RvNN being used for parsing a [10] describes a greedy method by which selecting the correct tree can be used as a neural network training criterion. In the Syntactically Untied RvNN (**SU RvNN**)[11], rather than reusing a single weight matrix for all merges a different weight matrix is used depending upon the the suggested class of the constituents (Eg Noun Phrase vs Verb Phrase), thus giving the model more power to represent the differing relationships.

The Matrix Vector RvNN (**MV RvNN**) also extends the RvNN to give better capacity for representing relationships[12]. In this model rather than each embedding just being a vector, each is a vector paired with a matrix, where the matrix is used to transform the other vector during the merge step. Thus every word is a relation.

The Recursive Neural Tensor Network (**RvNTN**) [16] takes the approach of the RvNN and applies it to the Neural Tensor Network (**NTN**)[15]. Once again the goal is to increase the capacity of the model to encode relationships. But rather than having many many matrices a single tensor (higher order matrix) is used.

The Dependency Tree RvNN (**DT RvNN**)[4], use a dependency tree rather than a consistency tree as was used in the original RvNN. It is believed that the dependency tree is more invariant to syntactic changes. The Semantic variant of the DT RvNN uses the dependency matrix, rather than the positional matrix from the parse to encode the relationship of the a node to its many children[4]. The Dependency Tree Recursive Autoencoder (**DT RAE**) is the the corresponding autoencoder for a dependency tree[14].

B.2.2. Sentiment Analysis

Sentiment Analysis is the most commonly used technique to evaluated sentence embeddings today. Results for the basic Polarity classification task are shown in Table 2, and for the more challenging Exact Sentiment Analysis task are shown in Table 3.

Recursive Autoencoders The Recursive AutoEncoder (**RAE**) is the application of an RvNN to autoencoding [7]. This is a unsupervised task where the model simply has to learn to reconstruct is input. It is valuable as it lets the model learn structures from the very large amounts of unlabeled training data. In the original RAE definition given in [7] the task is to minimize the reconstruction error for all merges. The Unfolding Recursive Autoencoder (**URAE**) extends up this to used the better loss function of minimizing the reconstruction error at the word level only – which is the only error that truly matters. The Bilingual Recursive Autoencoder (**BRAE**), is formed by creating two URAEs for different languages, then linking their central embedding layer to provide an additional supervised criterion of sentenced with the same meaning in the two networks having the same position in the vector space – thus creating a common space between the languages allowing the measurement of how similar statements are.

Year	Author	Method	Data	Performance
2011	Socher et. al.[7]	RAE	Movie Reviews[25]	Acc: 77.7%
			MPQA Opinions	Acc: 86.4%
2013	Socher et. al. [16]	Recursive NTN	Sentiment Treebank	Acc: 87.6%
2014	Kalchbrenner et. al.[18]	Dynamic CNN		Acc: 86.8%
2014	Le and Mikolov[23]	PV-DBOW + PV-DM		Acc: 87.8%
2014			IMDB Dataset	Acc: 92.6%
2015	Zhang and LeCun[21]	Temporal CNN on characters	Amazon Reviews	Acc: 95.1%

Table 2: The application of various model to the Polarity Sentiment Analysis task. For this task a correct results is limited to determining if the statement is negative or positive.

Year	Author	Method	Data	Performance
2013	Socher et. al. [16]	Recursive NTN	Sentiment Treebank	Acc: 80.7%
2014	Le and Mikolov[23]	PV-DBOW + PV-DM		Acc: 48.7%
2014	Kalchbrenner et. al.[18]	Dynamic CNN		Acc: 48.5%
2015	Zhang and LeCun[21]	Temporal CNN on characters	Amazon Reviews	Acc: 59.57%
2012	Socher et. al.[12]	MV-RvNN	Movie Reviews[25]	Acc: 79.0%

Table 3: The application of various model to the Exact Rating Sentiment Analysis task. For this task a correct results is determining the exact rating the reviewer gave accompanying the review statement.

While these Autoencoders sound like they are usable for resynthesis, they are unfortunately not. During the reconstruction of all models discussed, they need to be provided with the tree structure of the output. If the tree structure of the output is not known – e.g. if the vector was generated as the centroid of a cluster, as is shown in Figure B.1 then they can not be used. The DT RAE of [14] does show proof of concept for using arbitrary trees for reconstruction, though no quantitative evaluation was carried out, nor was method provided for choosing the optimal tree.

Paragraph Vector Models [23] presents two new Paragraph Vector models. The name may be misleading, they are applicable to sequences of words of any length, including sentences. Both models are extensions of worded embedding models, which are themselves extensions of the original word embedding paper[22] on Neural Network Language

Models (**NP-LM**). The NP-LM is a model tasked with predicting the next word given the previous words. It looks at a fixed length window of words at a time. The inputs to the neural network are vectors, initially random retrieved from a look up table of words, then trained during network draining – just like in the RvNN family.

The Continuous Bag of Words (**CBOW**) is an optimized form of the NP-LM, using techniques like hierarchical soft-max[26], and averaging rather than concatenating the inputs within the window. The Distributed Memory Paragraph Vector (**PV DM**)[23], extends on this and the NP-LM, by giving an additional input to all the windows in the same sentence/paragraph, the "paragraph vector".

The Skip nGram, word embedding model extends and to an extent inverts the NP-LM, by instead tasking a single word vector and tasking the network to output its adjacent words. The Distributed Bag of Words Paragraph Vector (**PV DBOW**) method, modified the Skip nGram this to function with paragraphs by replacing the word-vector inputs with paragraph vectors.

Convolutional Neural Networks The final family of neural networks for NLP reviewed here are the Convolution Neural Networks (**CNN**s). CNNs use convolution and pooling layers to force structure on to deep neural networks[19, 20]. They most well known for achieving unparalleled results on image recognition tasks. The work of [21] applies a CNN directly to character data – unlike all other methods here it considers sentences not as strings of words but as strings of characters. Max-Pooling is used to solve the problem of the inputs having different lengths. It is a very impressive results – accomplishing NLP tasks with no real prior knowledge.

The Dynamic CNN (**DCNN**)[18], does work over word embeddings, combining them with a Time Delay Neural Network (**TD NN**)[17]. The creators of the DCNN suggest that the network is learning trees like the RvNN family within the CNN.

B.2.3. Scholars in the Field

- Dr Fei Liu, School of Computer Science, Carnegie Mellon University, USA. Email: feiliu@cs@cmu.edu[1]

- A/Prof. Phil Blunsom, Department of Computer Science, University of Oxford. UK. Email: phil.blunsom@cs.ox.ac.uk

- Dr Richard Socher, Computer Science Department,Stanford University, USA. Email: richard@socher.org

- Dr Tomas Mikolov, Facebook AI Research, USA. Email: tmikolov@fb.com

- Prof. Yann LeCun, Computer Science Department, New Your University, USA. Email: yann@cs.nyu.edu

[1]Dr Fei Liu will be moving to the University of Central Florida in the very near future. Her future contact details are thus expected to change.

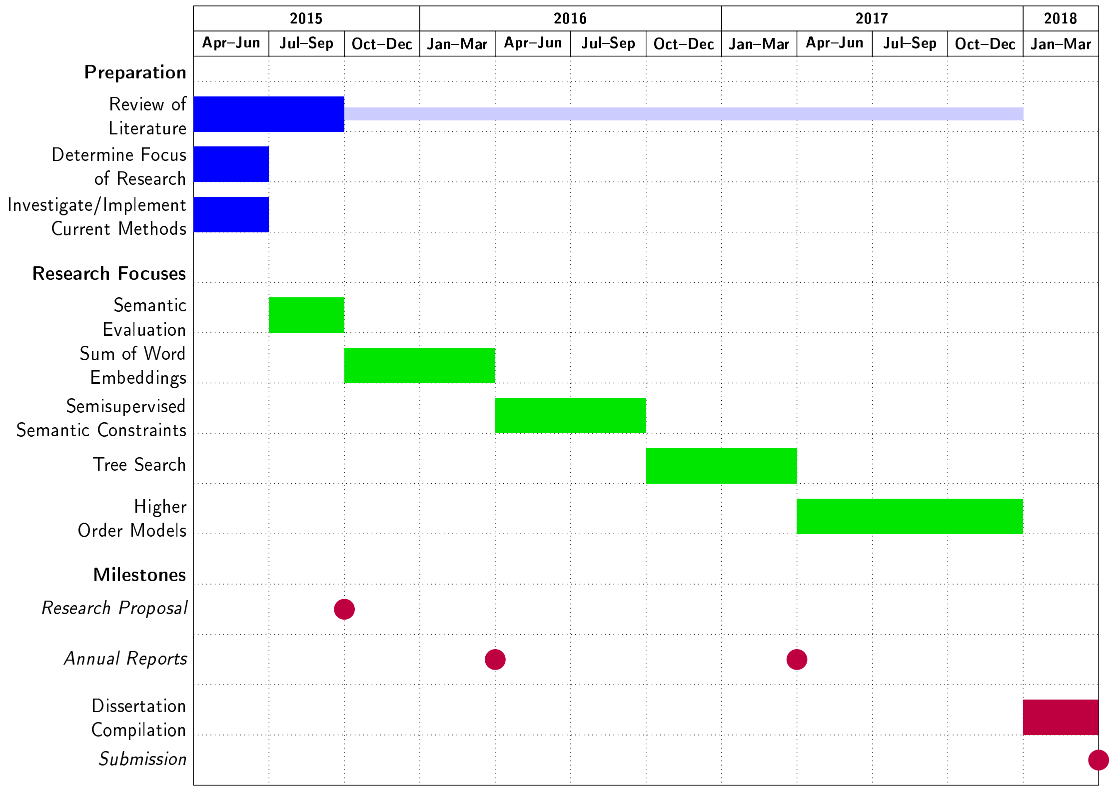

Figure B.3: An overview of the research program proposed.

- Prof. Yoshua Bengio, Department of Computer Science and Operations Research, Canada, Email: yoshua.bengio@umontreal.ca

B.2.4. Project Plan

The timeline for this project is broken up into sections, as shown in Figure B.3. Further details on the timeline be found in section §D, and in the Confirmation of Candidature documentation attached.

Work Completed: Semantic Evaluation Current methods for assessing the quality of representations are not sufficient to a certain whether they are sufficiently semantically consistent for use in resynthesis. The predominant assessment method is the use as a classifier for sentiment analysis. While sentiment analysis does have semantic component, it is far too loosely grained to reveal if sentences with the same meaning are collocated in vector space. Thus a new method for direct semantic evaluation was created. To enable the checking of the semantic localization sentences from real world corpora were categorized into groups of paraphrases – sentences with the same semantic meaning. They groups were stratified into testing and training splits and a linear support vector machine was used to classify the sentences in their paraphrase groups. The SVM was given embeddings from PV-DM, DBOW, URAE, and the Mean of Word Embeddings each in turn. Being able to correctly classify the sentences indicate that the embeddings and meanings consistently align and do not overlap. Unexpectedly, the Mean of Word Embeddings performed substantially better than the more complex models. This result does line-up with the recent result of [27] for Sum of Word Embeddings.

Sum Of Word Embeddings A sentence as the sum of its words is a neat, and unexpected result. It bears investigation and development. Resynthesizing a sentence from it's sum is a theoretically intractable problem, but with suitable approximations and heuristics may be feasible. The problem of working out which word embeddings add up to a particular vector is equivalent to the Subset Sum problem which is proven NP-Complete. The second problem is the loss of word order. Both may be solved through the use of classical statistical NLP: collecting co-occurrence, and n-gram statistics from a suitable corpus. The co-occurrence statics can be used to greedily prune the search space so that embeddings words which are only rarely found together are not considered as possible partial sums. The ngrams can be used to order the words, once the unordered set is solved for. The problem would also require aggressive dimensionality reduction to reduce to time-complexity. Preliminary results have shown that it is viable to preform substantial dimensionality reduction, without correspondent loss of information. This is helped, as it the overall problem, by restricting the vocabulary – for example only using the 1000 most common English words. Given these constraints and heuristics, it should be feasible to tackle the problem using dynamic programming, or ant-colony optimization.

Other investigations to be co-occurring with the development of resynthesis, would be extending word embeddings to use the syntactic parts of speech labels. Current word embeddings are based on only the word – that means that there is one embedding for bank as in to "bank a plane", as for bank as in "the bank of a river". Using current systems it is possible to reliably parts of speech tag, which will allow the separation of such homonyms. However it does not completely solve the problem, A financial "bank" and a river bank are still the same. A complete solution would be require word sense disambiguation. Currently, the best word sense disambiguation algorithms barely perform above always choose the most common word sense. This is worse than assigning a combined embedding for all uses with position given as a frequency occurrence weighted centeroid of the true positions.

Semi-supervised Semantic Constraints Drawing on the semantic requirements determined in the Semantic Evaluation subproject, in this section is is proposed to optimize for those constraints directly. Forcing the embeddings to be collocated with other embeddings for sentences of the same meaning. As in the Bilingual Recursive Autoencoder (**BRAE**), two connect two Unfolding Recursive Autoencoders (**URAE**s) will be connected at the central merge, and an error signal created based on the difference in position of two sentences which are deemed semantically equivalent. Unlike in the BRAE, both URAEs will be for the same language (nominally English), and will infact be the same URAE but with different inputs. The supervised error signal from the difference in position will supplement the unsupervised reconstruction error signal, thus forming a semisupervised model.

In preparing this model several other techniques will also be brought to bare. The enhanced parts-of-speech word vectors developed in the previous section may be used. Also, rather than the traditional constituency tree URAE, a Dependency Tree URAE (**DT URAE**) as in the work of [13] may be used. Through these methods semantically consistent recursive embeddings will be created.

Tree Search Given semantically consistent embeddings from a compositional (tree) model like an RAE, reconstructing a corresponding sentence is a matter of finding the right tree to decompose it into. [13] shows a proof of concept for this task, though they do not propose an algorithm for selecting the tree, just how it can be done if the tree is given. A method for selecting the tree with be developed in this subproject. A naive algorithm, as a starting point, would be selecting the tree which for which the leaf-nodes are closest to existent word embeddings. Such an algorithm would require some cut-offs to avoid searching the infinite space of all possible trees. The issues are not dissimilar to those encountered when reconstructing from Sum of Word Embeddings, thus some techniques may be transferred.

Higher Order Models The preceding subprojects have been based around traditional neural networks, albeit with nontraditional topology. In this subproject it is proposed to tackle the same problems using higher order models, which can encode more complex relationships. The Matrix-Vector RvNN (**MV-RvNN**) [12], the Neural Tensor Network (**NTN**) [15] and the Recursive NTN (**RvNTN**)[16] are three such models. to the candidates knowledge, none have been applied as an autoencoder. Other models such as the Deep Tensor Neural Network (**DTNN**) [28] and Tensor Deep Stacking Network (**T-DSN**)[29] have been successfully applied to to speech recognition tasks, though not yet to the related NLP tasks. In this subproject such models will be investigated.

References

[1] M. Kågebäck, O. Mogren, N. Tahmasebi, and D. Dubhashi, "Extractive summarization using continuous vector space models," in *Proceedings of the 2nd Workshop on Continuous Vector Space Models and their Compositionality (CVSC)@ EACL*, 2014, pp. 31–39. [Online]. Available: http://www.aclweb.org/anthology/W14-1504

[2] D. Yogatama, F. Liu, and N. A. Smith, "Extractive summarization by maximizing semantic volume," *Conference on Empirical Methods in Natural Language Processing*, 2015. [Online]. Available: http://www.cs.cmu.edu/~dyogatam/papers/yogatama+liu+smith.emnlp2015.pdf

[3] R. Socher, E. H. Huang, J. Pennington, A. Y. Ng, and C. D. Manning, "Dynamic pooling and unfolding recursive autoencoders for paraphrase detection," in *Advances in Neural Information Processing Systems 24*, 2011. [Online]. Available: http://www.socher.org/uploads/Main/SocherHuangPenningtonNgManning_NIPS2011.pdf

[4] R. Socher, A. Karpathy, Q. V. Le, C. D. Manning, and A. Y. Ng, "Grounded compositional semantics for finding and describing images with sentences," *Transactions of the Association for Computational Linguistics*, vol. 2, pp. 207–218, 2014. [Online]. Available: https://tacl2013.cs.columbia.edu/ojs/index.php/tacl/article/download/325/45

[5] F. Rosenblatt, "Principles of neurodynamics. perceptrons and the theory of brain mechanisms," DTIC Document, Tech. Rep., 1961.

[6] D. E. Rumelhart, G. E. Hintont, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, p. 9, 1986. [Online]. Available: http://www.cs.toronto.edu/~hinton/absps/naturebp.pdf

[7] R. Socher, J. Pennington, E. H. Huang, A. Y. Ng, and C. D. Manning, "Semi-supervised recursive autoencoders for predicting sentiment distributions," in *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2011. [Online]. Available: http://www.socher.org/uploads/Main/SocherPenningtonHuangNgManning_EMNLP2011.pdf

[8] J. Zhang, S. Liu, M. Li, M. Zhou, and C. Zong, "Bilingually-constrained phrase embeddings for machine translation." ACL, June 2014. [Online]. Available: http://anthology.aclweb.org/P/P14/P14-1011.pdf

[9] J. B. Pollack, "Recursive distributed representations," *Artificial Intelligence*, vol. 46, no. 1â"2, pp. 77 – 105, 1990. [Online]. Available: http://www.sciencedirect.com/science/article/pii/000437029090005K

[10] R. Socher, C. C. Lin, C. Manning, and A. Y. Ng, "Parsing natural scenes and natural language with recursive neural networks," in *Proceedings of the 28th international conference on machine learning (ICML-11)*, 2011, pp. 129–136. [Online]. Available: http://nlp.stanford.edu/pubs/SocherLinNgManning_ICML2011.pdf

[11] R. Socher, J. Bauer, C. D. Manning, and A. Y. Ng, "Parsing with compositional vector grammars," in *ACL*, 2013. [Online]. Available: http://www.socher.org/uploads/Main/SocherBauerManningNg_ACL2013.pdf

[12] R. Socher, B. Huval, C. D. Manning, and A. Y. Ng, "Semantic compositionality through recursive matrix-vector spaces," in *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*. Association for Computational Linguistics, 2012, pp. 1201–1211. [Online]. Available: http://www.aclweb.org/anthology/D12-1110

[13] M. Iyyer, J. Boyd-Graber, and H. D. III, "Generating sentences from semantic vector space representations," in *NIPS Workshop on Learning Semantics*, 2014. [Online]. Available: http://cs.umd.edu/~miyyer/pubs/2014_nips_generation.pdf

[14] M. Iyyer, J. Boyd-Graber, L. Claudino, R. Socher, and H. Daumé III, "A neural network for factoid question answering over paragraphs," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 633–644. [Online]. Available: https://cs.umd.edu/~miyyer/pubs/2014_qb_rnn.pdf

[15] R. Socher, D. Chen, C. D. Manning, and A. Y. Ng, "Reasoning with neural tensor networks for knowledge base completion," in *Advances in Neural Information Processing Systems 26*, 2013. [Online]. Available: http://nlp.stanford.edu/~socherr/SocherChenManningNg_NIPS2013.pdf

[16] R. Socher, A. Perelygin, J. Y. Wu, J. Chuang, C. D. Manning, A. Y. Ng, and C. Potts, "Recursive deep models for semantic compositionality over a sentiment treebank," in *Proceedings of the conference on empirical methods in natural language processing (EMNLP)*, vol. 1631. Citeseer, 2013, p. 1642. [Online]. Available: nlp.stanford.edu/~socherr/EMNLP2013_RNTN.pdf

[17] A. Waibel, T. Hanazawa, G. Hinton, K. Shikano, and K. J. Lang, "Phoneme recognition using time-delay neural networks," *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 37, no. 3, pp. 328–339, 1989. [Online]. Available: http://www.cs.toronto.edu/~fritz/absps/waibelTDNN.pdf

[18] N. Kalchbrenner, E. Grefenstette, and P. Blunsom, "A convolutional neural network for modelling sentences," *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, vol. Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, 2014. [Online]. Available: http://nal.co/papers/Kalchbrenner_DCNN_ACL14

[19] L. E. Atlas, T. Homma, and R. J. Marks II, "An artificial neural network for spatio-temporal bipolar patterns: Application to phoneme classification," in *Proc. Neural Information Processing Systems (NIPS)*, 1988, p. 31. [Online]. Available: http://goo.gl/4OApDV

[20] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998. [Online]. Available: http://yann.lecun.com/exdb/publis/pdf/lecun-01a.pdf

[21] X. Zhang and Y. LeCun, "Text understanding from scratch," *CoRR*, vol. Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, 2015. [Online]. Available: http://arxiv.org/abs/1502.01710

[22] Y. Bengio, H. Schwenk, J.-S. Senécal, F. Morin, and J.-L. Gauvain, "Neural probabilistic language models," in *Innovations in Machine Learning*. Springer, 2006, pp. 137–186. [Online]. Available: http://www.jmlr.org/papers/volume3/bengio03a/bengio03a.pdf

[23] Q. Le and T. Mikolov, "Distributed representations of sentences and documents," in *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, 2014, pp. 1188–1196. [Online]. Available: http://jmlr.org/proceedings/papers/v32/le14.pdf

[24] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013. [Online]. Available: http://arxiv.org/pdf/1301.3781v3

[25] B. Pang and L. Lee, "Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales," in *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*. Association for Computational Linguistics, 2005, pp. 115–124. [Online]. Available: http://ssli.ee.washington.edu/conferences/ACL2005/ACL/pdf/ACL15.pdf

[26] F. Morin and Y. Bengio, "Hierarchical probabilistic neural network language model," in *Proceedings of the international workshop on artificial intelligence and statistics*. Citeseer, 2005, pp. 246–252. [Online]. Available: http://www.iro.umontreal.ca/~lisa/pointeurs/hierarchical-nnlm-aistats05.pdf

[27] S. Ritter, C. Long, D. Paperno, M. Baroni, M. Botvinick, and A. Goldberg, "Leveraging preposition ambiguity to assess compositional distributional models of semantics," *The Fourth Joint Conference on Lexical and Computational Semantics*, 2015. [Online]. Available: http://clic.cimec.unitn.it/marco/publications/ritter-etal-prepositions-starsem-2015.pdf

[28] D. Yu, L. Deng, and F. Seide, “The deep tensor neural network with applications to large vocabulary speech recognition,” *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 21, no. 2, pp. 388–396, 2013. [Online]. Available: http://www.msr-waypoint.net/pubs/177443/DTNN-TASLP2012-Proof.pdf

[29] B. Hutchinson, L. Deng, and D. Yu, “Tensor deep stacking networks,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 35, no. 8, pp. 1944–1957, 2013.

C. Research Project Details

- This project does not involve the collection of confidential or sensitive information.
- The are no current intellectual property agreements relating to this research. There are no plans to commercialize the products of this research during the duration of the candidature.
- This project does not involve fieldwork.
- Additional computational resources are required and have been acquired for the project. For purposes of experimentation and development of the algorithms distributed computing methods are used. The estimated requirements are 32 CPU-Cores and at least 2Gb RAM per core. Further more. the computers need to be collocated on a highspeed network. A successful application has been made to NeCTAR for this allocation. The allocation will need to be renewed on the 31st of December 2015.
- There are no advanced statistical analysis required for this project, beyond the consideration and design of the algorithms developed as statistical analysis tools themselves.
- The outputs of the project will be communicated by publication in journals and conferences. As is conventional within this area, research will be primarily communicated though conference papers. The intent is to publish one conference paper in the first year, two in the second (targeting the same conference), and one in the third. As judged appropriate based on research in question, one or more of these conference papers will be extended into a journal article. The thesis shall be presented as this series of conference and journal papers, with additional introductory and concluding chapters.
- This project does not require any approvals. It has been investigated if the intended research publications require Defense Export Control Office (DECO) approval for publications. DECO approval is not required for publication.
- Any new and derived data sets will be placed in the Signals and Information Processing lab’s existing UWA Institutional Research Data Store (IRDS). Where licensing allows it will also be publicly published though the candidates website. The focus of this course of study is to create new methods of processing data rather

than new data. The methods for processing data will be version controlled via private Github, with intent to open source them at the pertinent times.

In the table below is shown a skills audit for the skills required in this project

Professional and Research Skills	Rating				Evidence	Plan for Acquisition
	None	Basic	Competent	Proficient		
Understanding and application of data collection and analysis methods			C		Completion of Honours project, which involved collection of large amounts of data, and its analysis.	Not Required
Identifying and accessing appropriate bibliographic resources			C		Annotated bibliography maintained. Completed Honours project. Completed UWA Library "Keeping Up to Date" workshop	Not Required
Understanding of mathematics required for this area (Probability, Linear Algebra)			C		Completed Pure Mathematics Major, as part of BCM	Not Required.
Use of programming languages for this area (Matlab, Python, Julia)				P	Completed Computation Major, as part of BCM Experience as professional software developer	Not Required
Use of signal processing techniques			C		Completed Electrical and Electronic Program as part of BE	Not Required
Use of Distributed Computing Resources (MPI etc)			C		Completed Developer Training at Pawsey Super Computer Center.	Complete
Principles and conventions of academic writing		B			Completion of Honours. However, this took intensive editing.	Attend GRS and Library Writing Workshops
Self discipline and motivation			C		Have worked at lower paying, much less enjoyable jobs to get to university.	Not Required

Time and project management			C		Completion of Honours. Completion of heavily project assessed Computer Science and Engineering Majors Including 4 project management units.	Not required
Awareness of issues relating to intellectual rights			C		Attended Graduate Research School Induction Session on Scholarly Ethics. Read the UWA Code of Ethics.	Not Required
Ability to constructively defend research outcomes at presentations			C		Have presented my Honours at school symposium. Have presented school seminar.	Not Required

D. Timeline: Research Training and Academic Tasks

The timeline for this research program is spread over 3 years, to inline with the candidate's Australia Post Graduate Award (APA) duration. If particular difficulties arise, the APA can have a 6 month extension, this also is indicated in the timeline below, to allow for adjustment to be scaled. Failure to complete before the termination of the funding will result in sever difficulties to the candidate's living circumstance and will likely result in non-completion.

This timeline should be reach in conjunction with the Candidature Tasks, on the cover-sheet, and the Research Program Overview in the at the end of this document. Key milestones are marked in *italics*.

01/03/2009 Academic Conduct Essentials (candidature task)

08/03/2015 *Enrollment*

24/04/2015 GRS: Theses and Publications Seminar

04/06/2015 GRS: Research Skills Workshop

05/06/2015 Pawsey Supercomputer Training (candidature task)

22/06/2015 GRS: How to Write a Research Proposal (at weekend writing retreat)

16/07/2015 Library Workshop: Keeping Up to Date [with Literature]

18/09/2015 *Research proposal*

08/03/2016 *Annual report Year 1*

08/03/2016 *Confirmation of candidature*

Description	Costs			Source	
	Year 1	Year 2	Year 3	School	GRS
Administrative and Research Costs					
Workstation	$1500	$0	$0	$1500	$0
Linguistic Data Consortium Membership	$0	$2400	$0	$2400	$0
Training Costs					
GRS/Library Seminars and Workshops	$0	$0	$0	$0	$0
Statistics Training Course	$0	$198	$0	$198	$0
Conference Attendance					
Domestic: Flights, Registration, Accommodation	$1500	$0	$1500	$3000	$0
International: Flights, Registration, Accommodation	$0	$2000	$0	$150	$1850
Subtotal:	$3000	$4598	$1500	$7248	$1850
				Total:	$9098

Table 5: Budget

15/02/2017 *Nomination of Examiners*

08/03/2017 *Annual report Year 2*

08/02/2018 Dissertation Draft submitted to supervisors

08/03/2018 *Dissertation submitted for examination*

08/03/2018 APA Termination

05/09/2018 APA Extension Termination

E. Budget

The budget for this research program is detailed in Table 5. The most significant cost of the project is the purchasing of a 2016 membership to the Linguistic Data Consortium. This membership allows the obtaining of the vast majority of the LDC data-sets at no additional cost. The piece-wise cost for the key data sets required for this research, Gigaword v5 and Penn Treebank-3, would otherwise cost $6,000 and $1,500 respectively. As this is an institution wide membership, it will also allow the group to obtain and update many other data-sets used for other projects. It was determined to obtain membership in the second year of the project, rather than the first to ensure best utilization.

F. Supervision

Principal & Coordinating Supervisor: Professor Roberto Togneri (40%)

- Provide expertise in spoken language systems, statistical signal processing and pattern recognition.

- Directing overall research training program
- Reviewing research outputs
- Provide regular feedback, on both overall, and current subproject progress

Co-Supervisor: Dr Wei Liu (40%)

- Provide expertise in natural language processing, and the conventions of publication in the field.
- Reviewing research outputs
- Provide regular feedback, on both overall, and current subproject progress

Co-Supervisor: Winthrop Professor Mohammed Bennamoun (20%)

- Provide expertise in machine learning, particularly in deep neural systems
- Reviewing research outputs
- Provide regular feedback, on both overall, and current subproject progress