**Copy from:** [**https://livebook.manning.com/book/transfer-learning-for-natural-language-processing/chapter-1/v-2/74**](https://livebook.manning.com/book/transfer-learning-for-natural-language-processing/chapter-1/v-2/74)

**1.3 A Brief History of NLP Advances**

In order to frame your understanding of the state and importance of Transfer Learning in NLP, it can be helpful to first gain a better sense of the kinds of tasks and techniques that have historically been important for this subfield of AI. That is the route taken by this section, which culminates in a brief overview of recent advances in NLP Transfer Learning. This will help you appropriately contextualize the impact of Transfer Learning in NLP, and understand why it is more important now than ever before.

**1.3.1 General Overview**

NLP was born in the middle of the 20th century, alongside AI. The first major historical NLP landmark was the Georgetown Experiment of 1954, where a set of approximately sixty Russian sentences were translated into English. In the 1960s, the Massachusetts Institute of Technology (MIT) system ELIZA convincingly simulated a psychotherapist. Also in the 1960s, the vector space model for information representation was developed, where words came to be represented by vectors of real numbers which were amenable to computation. The 1970s saw the development of a number of chatterbot/chatbot concepts based on sophisticated sets of hand-crafted rules for processing the input information.

The 1980s and 1990s saw the advent of the application of systematic machine learning methodologies to NLP, where rules were discovered by the computer versus being crafted by humans. This coincided with the explosion in the wider popularity of machine learning during that time, as we have already discussed earlier in this chapter. Late 1980s witnessed the application of singular value decomposition (SVD) to the vector space model, leading to latent semantic analysis – an unsupervised technique for determining the relationship between words in language.

In the early 2010s, the rise of neural networks and deep learning in the field dramatically transformed NLP. Such techniques were shown to achieve state-of-the-art results for the most difficult NLP tasks, such as machine translation and text classification. Mid 2010s witnessed the development of the word2vec model (Mikolov, Chen, Corrado, & Dean, 2013), and its variants sent2vec (Pagliardini, Gupta, & Jaggi, 2018)· , doc2vec (Le & Mikolov, 2014)· , etc.

These neural-network-based techniques vectorize words, sentences, documents (respectively) in a way that ensures the distance between vectors in the generated vector space is representative of the difference in meaning between the corresponding entities, i.e., words, sentences, documents, etc. Indeed, some interesting properties of such embeddings allowed analogies to be handled – the distance between the words *Man* and *King* are approximately equal to the distance between the words *Woman* and *Queen* in the induced vector space, for instance. The metric used to train these neural-network-based models was derived from the field of linguistics, more specifically *distributional semantics*, and did not require labeled data – the meaning of a word was assumed to be tied to its context, i.e., the words surrounding it.

The variety of methods for embedding various units of text, i.e., words, sentences, paragraphs, documents, etc., became a key cornerstone of modern NLP. Once text samples are embedded into an appropriate vector space, analysis can often be reduced to the application of a well-known shallow statistical/machine learning technique for real vector manipulation, including clustering, classification, etc. This can be viewed as a form of *implicit transfer learning*, and a semi-supervised machine learning pipeline, since the embedding step is unsupervised and the learning step typically supervised. The unsupervised pre-training step essentially reduces the requirements for labeled data, and thereby computing resources required to achieve a given performance, something we will learn to leverage transfer learning to do for us for a broader range of scenarios in this book.

Around 2014, *sequence-to-sequence* models (Sutskever, Vinyals, & Le, 2014) were developed, and achieved a significant improvement in difficult tasks such as machine translation and automatic summarization. In particular, whereas pre-neural-network NLP pipelines would consist of several explicit steps, such as POS, dependency parsing, language modeling, etc., it was shown that machine translation could be carried out “sequence to sequence”. Here the various layers of a deep neural network automate all of these intermediate steps. These models learn to associate an input sequence, e.g., a source sentence in one language, with an output sequence, e.g. that sentence’s translation into another language – via an encoder that converts inputs into a context vector and a decoder that converts it into the target sequence. Both the encoder and decoder were typically designed to be Recurrent Neural Networks (RNNs). These are able to encode order information in the input sentence, something earlier models, such bag-of-words, couldn’t do, leading to significant improvements in performance.

**Attention** - It was however discovered that long input sequences were harder to deal with, which motivated the development of the technique known as *attention*. This significantly improved the performance of machine translation sequence-to-sequence models by allowing the model to focus on the parts of the input sequence that were most relevant for the output.

**Transformer** - A model called the **Transformer** (Vaswani et al., 2017) took this a step further by defining a self-attention layer for both the encoder and decoder, allowing both to build better context for text segments with respect to other text segments in the input sequence. Significant improvements in machine translation were achieved with this architecture, and notably it was observed to be better suited for training on massively parallel hardware than prior models, speeding up training by up to an order of magnitude.

Up to about 2015, most practical methods for NLP focused on the *word-level*. This just means that the whole word was treated as an indivisible atomic entity and assigned a feature vector. This approach has several disadvantages, notably how to treat never-before-seen or out-of-vocabulary words. When the model encountered such words, for instance if a word was misspelled, the method would fail as it could not vectorize it. Additionally, the rise of social media changed the definition of what was considered natural language. Now, billions of people came to express themselves online using emoticons, newly-invented slang, and frequently deliberately misspelled words. It was not long until it was realized that the solution to many of these issues came naturally from treating language at the character-level. In this paradigm, every character would be vectorized, and as long as the human was expressing themselves with allowable characters, vector features could be generated successfully and the algorithm could be successfully applied. Zhang et al. (Zhang, Zhao, & Lecun, 2015) showed this in the context of character-level CNNs for text classification, and demonstrated a remarkable robustness to misspellings.

**1.3.2 Recent Transfer Learning Advances**

Traditionally, learning has proceeded in either a **fully supervised** or **fully unsupervised** fashion for any given problem setting – a particular combination of task, domain and language – from scratch. As previously alluded to, **semi-supervised learning** was recognized as early as 1999, in the context of SVMs, as a way to address potentially limited labeled data availability. An initial unsupervised pre-training step on larger collections of unlabeled data made downstream supervised learning easier. Variants of this studied how to address potentially noisy, i.e., possibly incorrect, labels – an approach sometimes referred to as *weakly supervised learning*. However, it was often assumed that the same sampling distribution held for both the labeled and unlabeled datasets.

Transfer learning relaxes these assumptions. The need for transfer learning was arguably popularly recognized in 1995 – as the need for “Learning to Learn” at the 1995 edition of NeurIPS. This is probably the biggest conference in machine learning. Essentially, it stipulated that intelligent machines need to possess lifelong learning capabilities, which reuse learned knowledge for new tasks. It has since been studied under a few different names, including *learning to learn*, *knowledge transfer*, *inductive bias*, *multi-task learning*, etc. In multi-task learning, an algorithm is trained to perform well on multiple tasks simultaneously, thereby uncovering features that may be more generally useful. However, it wasn’t until around 2018 that practical and scalable methods were developed to achieve it in NLP for the hardest perceptual problems.

**Revolution -** The year 2018 saw nothing short of a revolution in the field of NLP. The understanding in the field of how to best represent collections of text as vectors evolved dramatically. Moreover, it became widely recognized that open-sourced models could be fine-tuned or transferred to different tasks, languages and domains. At the same time, several of the big Internet companies released even more and bigger NLP models for computing such representations, and also specified well-defined procedures for fine-tuning them.

All of a sudden, the ability to attain state-of-the-art results in NLP became accessible to the average practitioner, even an independent one. Some referred to this as NLP’s “ImageNet moment”, referencing the explosion in computer vision applications witnessed post 2012, when a GPU-trained neural networks won the ImageNet computer vision competition. Just as was the case for the original ImageNet moment, for the first time a library of pre-trained models became available for a large subset of arbitrary NLP data, together with well-defined techniques for fine-tuning them to particular tasks at hand with labeled datasets of size significantly smaller than would be needed otherwise. It is the purpose of this book to describe, elucidate, evaluate, demonstrably apply, compare and contrast the various techniques that fall into this category. We briefly overview these techniques next.

One notable weakness of the original formulation of word2vec was disambiguation. There was no way to distinguish between various uses of a word that may have different meanings depending on context, i.e., homographs, e.g., duck (posture) versus duck (bird), fair (a gathering) versus fair (just). In some sense, the original word2vec formulation represents each such word by the average vector of the vectors representing each of these distinct meanings of the homograph. *Embeddings from Language Models* (Peters et al., 2018) – abbreviated ELMo after the popular Sesame Street character – is an attempt to develop **contextualized embeddings of words**, using bidirectional LSTMs. The embedding of a word in this model depend very much on its context, with the corresponding numerical representation being different for each such context. ELMo did this by being trained to predict the next word in a sequence of words, which is very much related to the concept of language modeling that was introduced at the beginning of the chapter. Huge datasets, e.g., Wikipedia and various datasets of books, are readily available for training.

*Universal Language Model Fine-tuning* (Howard & Ruder, 2018) (ULM-Fit) is a method that was proposed to fine-tune any neural-network-based language model for any particular task, and was initially demonstrated in the context of **text classification**. A key concept behind this method is *discriminative fine-tuning*, where the different layers of the network are trained at different rates. The *OpenAI Transformer* modified the encoder-decoder architecture of the Transformer to achieve a fine-tunable language model for NLP. It discarded the encoders, retaining the decoders and their self-attention sublayers.

*Bidirectional Encoder Representations from Transformers* (Devlin, Chang, Lee, & Toutanova, 2019) (BERT) did arguably the opposite, modifying the **Transformer** architecture by preserving the encoders and discarding the decoders, also relying on *masking* of words which would then need to be predicted accurately as the training metric. These concepts will be discussed in detail in the upcoming chapters.

In all of these **language-model-based** methods – *ELMo*, *ULM-Fit*, the *OpenAI* Transformer and *BERT* – it was shown that embeddings generated could be fine-tuned for specific downstream NLP tasks with relatively few labeled data points. The focus on language models was deliberate - it was hypothesized that the hypothesis set induced by them would be generally useful, and the data for massive training was known to be readily available.

Beside language modeling, employing character-level Convolutional Neural Networks (CNNs) combined with bi-directional LSTMs for structural semantic text classification, the Semantic Inference for the Modeling of Ontologies (SIMOn)[· [13]](https://livebook.manning.com/book/transfer-learning-for-natural-language-processing/chapter-1/v-2/)[EK1] · approach, demonstrated NLP transfer learning methods directly analogous to those that have been used in computer vision. The rich body of knowledge on transfer learning for computer vision applications motivated this approach. The features learned by this model were shown to be also useful for unsupervised learning tasks, and to work well on social media language data which can be somewhat idiosyncratic and very different from the kind of language on Wikipedia and other large book-based datasets.

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). [BERT] 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, NAACLHLT 2019* (pp. 4171–4186). Minneapolis, MN, USA: Association for Computational Linguistics. Retrieved from http://arxiv.org/abs/1810.04805

Howard, J., & Ruder, S. (2018). [ULM-fit] 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)* (pp. 328–339). Stroudsburg, PA, USA: Association for Computational Linguistics. https://doi.org/10.18653/v1/P18-1031

Le, Q. V., & Mikolov, T. (2014). [Paragraph2Vec] [Doc2Vec] [seminal] Distributed Representations of Sentences and Documents. In *Proceedings of the 31st International Conference on Machine Learning* (Vol. 32, pp. 1188–1196). Beijing, China.

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). [Word2VecA] Efficient Estimation of Word Representations in Vector Space. In *1st International Conference on Learning Representations, ICLR 2013 - Workshop Track Proceedings*. Retrieved from http://arxiv.org/abs/1301.3781

Pagliardini, M., Gupta, P., & Jaggi, M. (2018). [sent2vec] Unsupervised Learning of Sentence Embeddings Using Compositional n-Gram Features. 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. 528–540). Stroudsburg, PA, USA: Association for Computational Linguistics. https://doi.org/10.18653/v1/N18-1049

Peters, M., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., & Zettlemoyer, L. (2018). [ELMo] Deep Contextualized Word Representations. 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. 2227–2237). Stroudsburg, PA, USA: Association for Computational Linguistics. https://doi.org/10.18653/v1/N18-1202

Sutskever, I., Vinyals, O., & Le, Q. V. (2014). [sequence-to-sequence] Sequence to Sequence Learning with Neural Networks. In *Advances in Neural Information Processing Systems*. Retrieved from http://arxiv.org/abs/1409.3215

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., … Polosukhin, I. (2017). [transformer] Attention Is All You Need. In *Advances in Neural Information Processing Systems (NIPS)* (pp. 5999–6009). Long Beach, CA, USA. Retrieved from http://arxiv.org/abs/1706.03762

Zhang, X., Zhao, J., & Lecun, Y. (2015). Character-level convolutional networks for text classification. In *Advances in Neural Information Processing Systems*.

[EK1]P. Azunre et al., “Semantic Classification of Tabular Datasets via Character-Level Convolutional Neural Networks”, Arxiv, 2019.