10/16/2023

Transfer Learning in Natural Language Processing

Aboli Gadekar

Abstract:

Transfer learning has revolutionized the field of natural language processing (NLP) by enabling models to leverage pre-trained knowledge on large datasets for various downstream tasks. This paper explores the fundamentals of transfer learning in NLP, its applications, and the key techniques that have shaped this domain. We also discuss the challenges and future directions in transfer learning for NLP.

Introduction:

We, humans, are very perfect at applying the transfer of knowledge between tasks. This means that whenever we encounter a new problem or a task, we recognize it and apply our relevant knowledge from our previous learning experiences. This makes our work easy and fasts to finish. For instance, if you know how to ride a bicycle and if you are asked to ride a motorbike which you have never done before. In such a case, our experience with a bicycle will come into play and handle tasks like balancing the bike, steering, etc. This will make things easier compared to a complete beginner. Such learnings are very useful in real life as it makes us more perfect and allows us to earn more experience.

Following the same approach, a term was introduced Transfer Learning in the field of machine learning. This approach involves the use of knowledge that was learned in some task and applying it to solve the problem in the related target task. While most machine learning is designed to address a single task, the development of algorithms that facilitate transfer learning is a topic of ongoing interest in the machine-learning community.

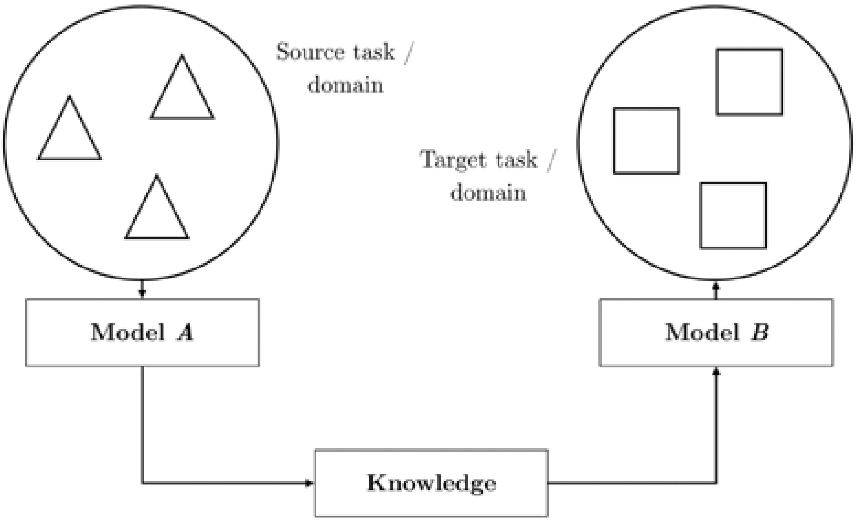


Fig 1: Transfer Learning Setup

Artificial intelligence (AI) has transformed modern society in a dramatic way. Tasks which were previously done by humans can now be done by machines faster, cheaper, and in some cases more effectively. Popular examples of this include computer vision applications concerned with teaching computers how to understand images and videos for the detection of criminals in closed-circuit television camera feeds, for instance. Other computer vision applications include detection of diseases from images of patient organs and the detection of plant species from plant leaves. Another important branch of AI, which deals particularly with the analysis and processing of human natural language data, is referred to as *natural language processing* (NLP). Examples of NLP applications include speech-to-text transcription and translation between various languages, among many others.

The embedding of a word in this model depends 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, a crucial task in the world of language modeling. Huge datasets, e.g. Wikipedia and various datasets of books, are readily available for training in this framework.

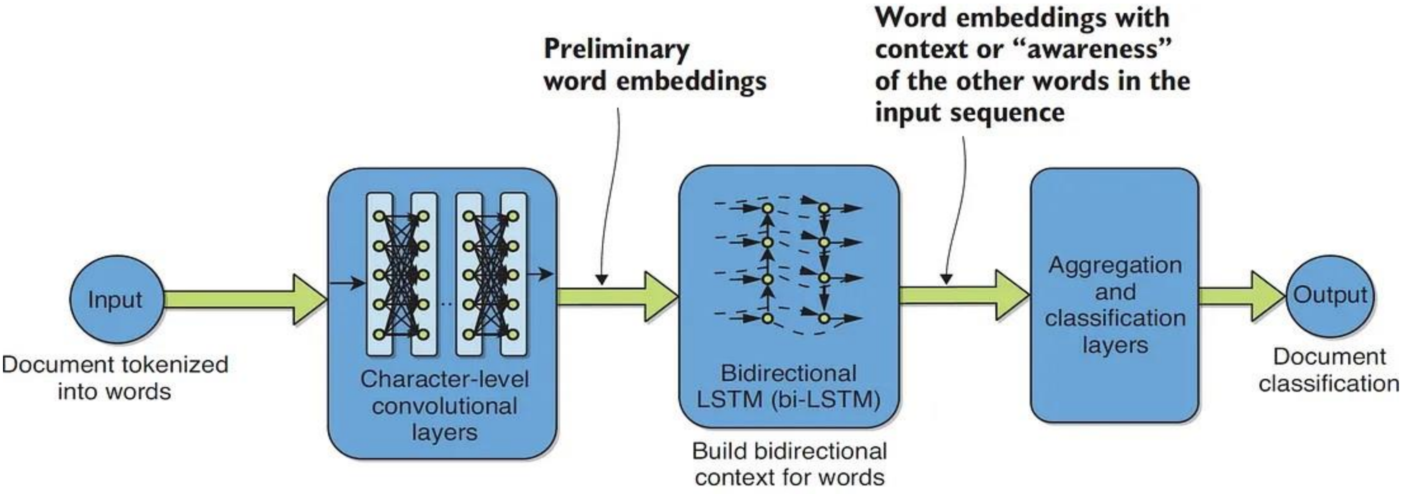

Preliminary word embeddings
Word embeddings with context or "awareness" of the other words in the input sequence
Input
Document tokenized into words
Character-level convolutional layers
Bidirectional LSTM (bi-LSTM)
Build bidirectional context for words
Aggregation and classification layers
Output
Document classification

Fig 2: Transfer Learning in NLP

The first major section of text in the paper, the Introduction commonly describes the topic under investigation, summarizes or discusses relevant prior research identifies unresolved issues that the current research will address, and provides an overview of the research that is to be described in greater detail in the sections to follow.

Background Study:

NLP stands for Natural Language Processing, which is a part of Computer Science, Human language, and Artificial Intelligence. It is the technology that is used by machines to understand, analyse, manipulate, and interpret human's languages. It helps developers to organize knowledge for performing tasks such as Translation, Automatic Summarization, Named Entity Recognition (NER), Speech Recognition, Relationship Extraction, and Topic Segmentation.

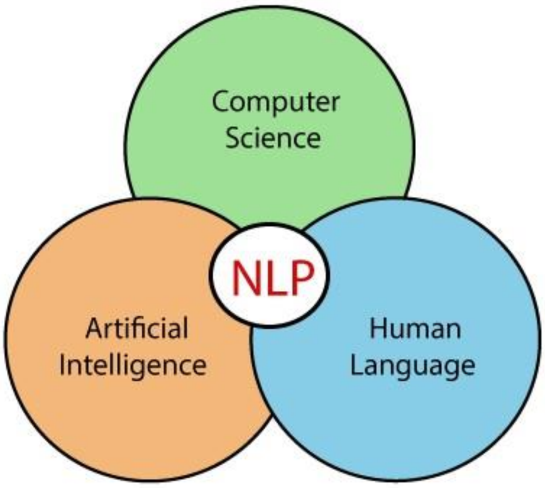


Fig 3: Natural Language Processing

Advantages of NLP:

- NLP helps users to ask questions about any subject and get a direct response within seconds.
- NLP offers exact answers to the question means it does not offer unnecessary and unwanted information.
- NLP helps computers to communicate with humans in their languages.
- Most of the companies use NLP to improve the efficiency of documentation processes, accuracy of documentation, and identify the information from large databases.

Disadvantages of NLP:

- It may not show context.
- It is unpredictable.
- It may require more keystrokes.
- It is unable to adapt to the new domain, and it has a limited function that's why NLP is built for a single and specific task only.

Components of NLP:

There are the following two components of NLP:-

Natural Language Understanding (NLU):

Natural Language Understanding (NLU) metadata from content such as concepts, entities, keywords, emotion, relations, and semantic helps the machine to understand and analyse human language by extracting the roles.

NLU mainly used in Business applications to understand the customer's problem in both spoken and written language.

NLU involves the following tasks:

- It is used to map the given input into useful representation.
- It is used to analyse different aspects of the language.

Natural Language Generation (NLG):

Natural Language Generation (NLG) acts as a translator that converts the computerized data into natural language representation. It mainly involves Text planning, Sentence planning, and Text Realization.

One of the main challenges of NLP is finding and collecting enough high-quality data to train and test your models. Data is the fuel of NLP, and without it, your models will not perform well or deliver accurate results. However, data is often scarce, noisy, incomplete, biased, or outdated.

Transfer Learning in NLP:

Pre-trained language models in NLP help us in doing exactly that and in the field of Deep Learning this idea is known as transfer learning. These models enable data scientists to work on a new problem by providing an existing model they can leverage to build upon to solve a target NLP task. Pre trained models have already proven its effectiveness in the field of Computer Vision. It has been a practice in Computer Vision to train models on the large image corpus such as Image Net that enabled the model to be better at learning the general image features such as curves and lines and then fine tune the model to the specific task. Due to the computational costs involved in training on such a large data set, the introduction of Pre Trained models came in as a boon for those who wanted to build their models accurately but faster and didn't want to spend time on training on the generic features needed for the task in question.

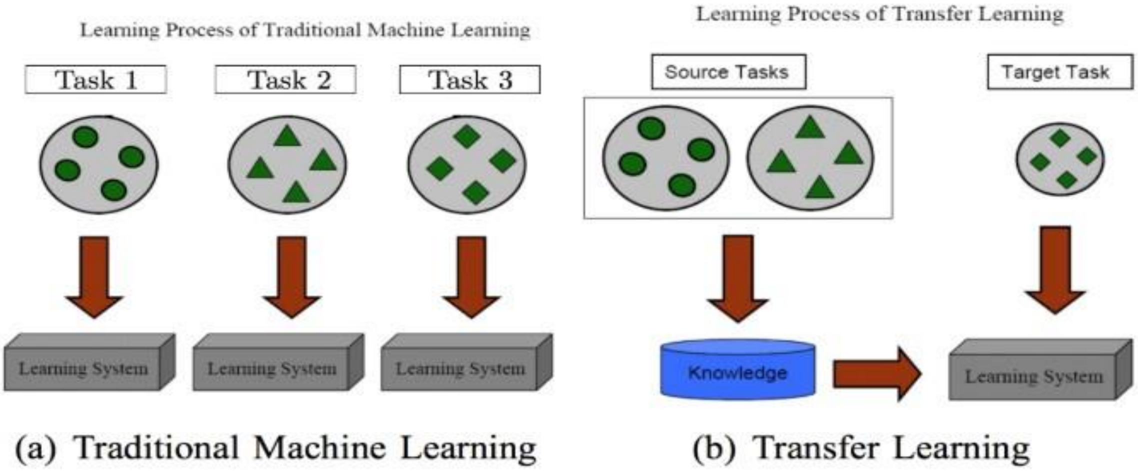
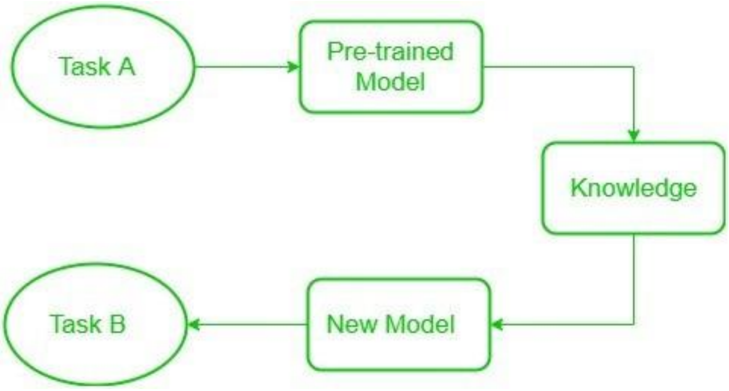


Fig 4: Difference between Traditional ML and Transfer Learning

Here's a simplified overview of how transfer learning works in NLP:

Pre-training:

In the pre-training phase, a neural network model is trained on a massive and diverse text corpus, often consisting of parts of the internet or large collections of books and articles. This pre-training phase helps the model to learn general language understanding and representation. Popular models used for pre-training include BERT (Bidirectional Encoder Representations from Transformers), GPT (Generative Pre-trained Transformer) and others.



Feature Extraction:

After pre-training, the model can be used for various NLP tasks. In some cases, you might extract features from the pre-trained model's layers and use them as input features for another model, such as a traditional machine learning classifier. These features are valuable because they encode a deep understanding of the language.

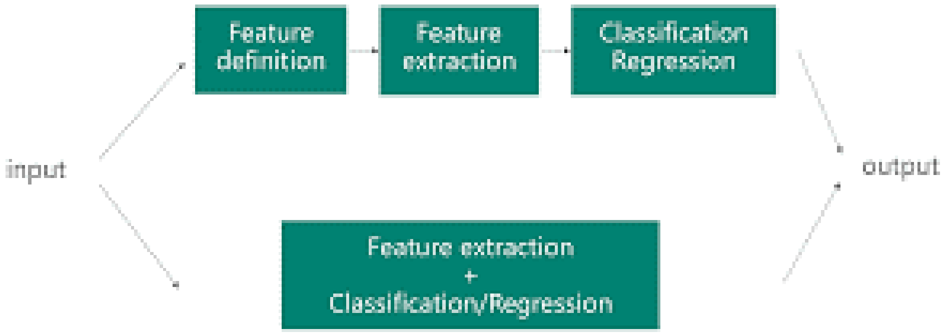


Fig 6: Feature Extraction

Fine-tuning:

In many transfer learning scenarios, fine-tuning is employed. This involves taking the pre-trained model and further training it on a smaller dataset specific to the target task. The model's architecture and parameters are adjusted slightly during this phase to adapt it to the specific task. Fine-tuning is crucial because it allows the model to specialize in the new task while retaining the general language understanding gained during pre-training.

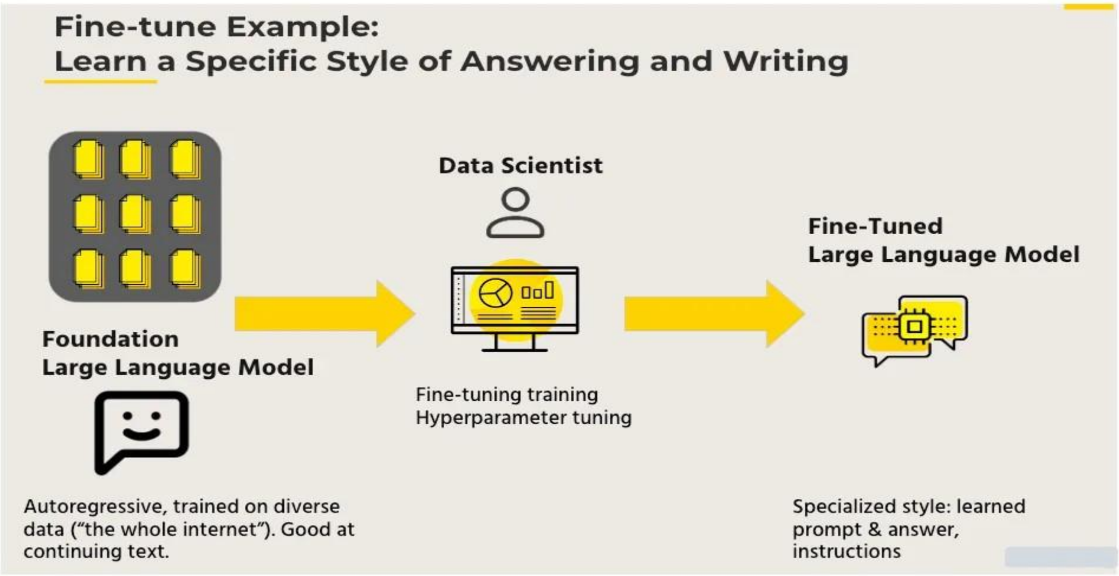

Fine-tune Example:
Learn a Specific Style of Answering and Writing
Data Scientist
Fine-Tuned
Large Language Model
Foundation
Large Language Model
Fine-tuning training
Hyperparameter tuning
Autoregressive, trained on diverse data ("the whole internet"). Good at continuing text.
Specialized style: learned prompt & answer, instructions

Fig 7: Fine-tuning

Transfer Learning Benefits:

- **Reduced Data Requirements:** Transfer learning enables models to perform well with smaller labeled datasets for specific tasks, which is especially useful when collecting labeled data is expensive or time-consuming.
- **Generalization:** Pre-trained models capture a wide range of linguistic patterns and general knowledge, making them effective for a variety of NLP tasks.
- **Faster Development:** Instead of building a model from scratch, you can start with a pre-trained model and fine-tune it for your specific task, saving time and resources.
- **State-of-the-Art Performance:** Many state-of-the-art NLP models and systems rely on transfer learning to achieve top performance on various benchmarks and tasks.

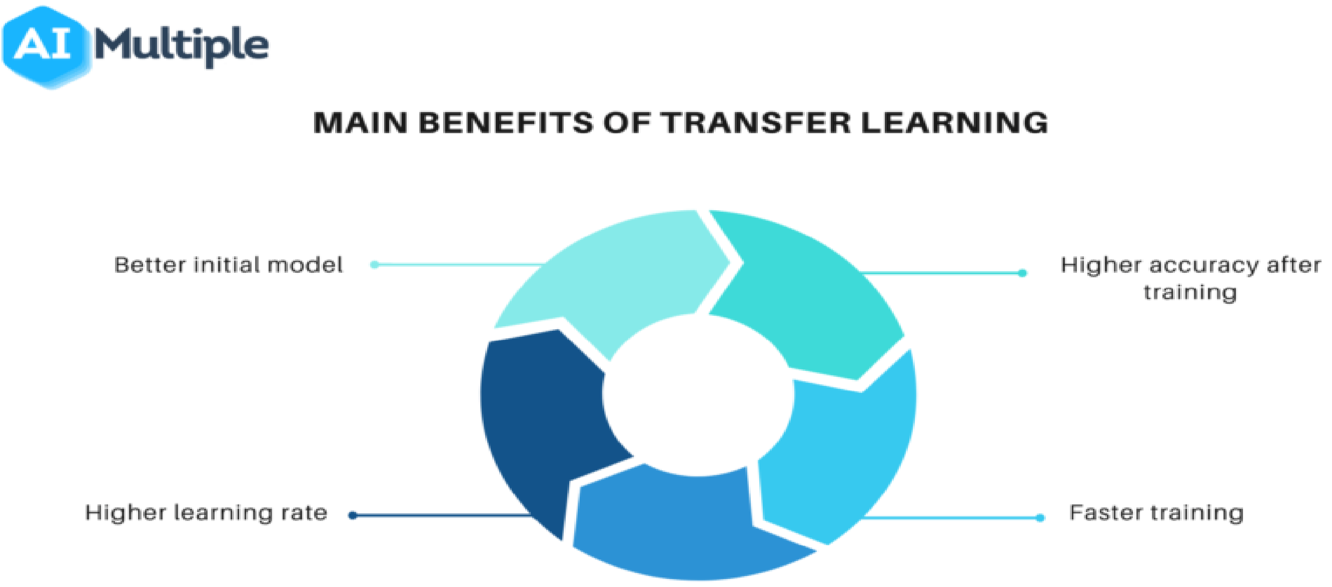


Fig 8: Advantages of Transfer Learning

Training machine learning models can be a challenging data science tasks. The training algorithms might not work as intended, training times can take too long, or training data can be problematic. Transfer learning is one of those techniques to make training easier. Just like humans can transfer their knowledge on one topic to a similar one, transfer learning can provide data scientists to transfer insights gained from a machine learning task into a similar one. By that, they can shorten machine learning model training time and rely on fewer data points.

Three advantages of transfer learning techniques:

- Transfer learning requires a small amount of local training data.
- In transfer learning, datasets for the fine-tuned model can be different from ones for the pre-trained model.
- Low requirement of computational resources.

Pre-trained Language Models:

Pre-trained models (PTMs) for natural language processing (NLP) are deep learning models, such as transformers, that have been trained on large datasets to perform specific NLP tasks. By training on extensive corpora, PTMs can learn universal language representations, which are useful for various downstream NLP tasks such as text summarization, named entity recognition, sentiment analysis, part-of-speech tagging, language translation, sentiment analysis, text generation, information retrieval, text clustering, and many more. This eliminates the need to train a new model from scratch each time. In other words, pre-trained models can be seen as reusable NLP models that developers can use to quickly build NLP applications.

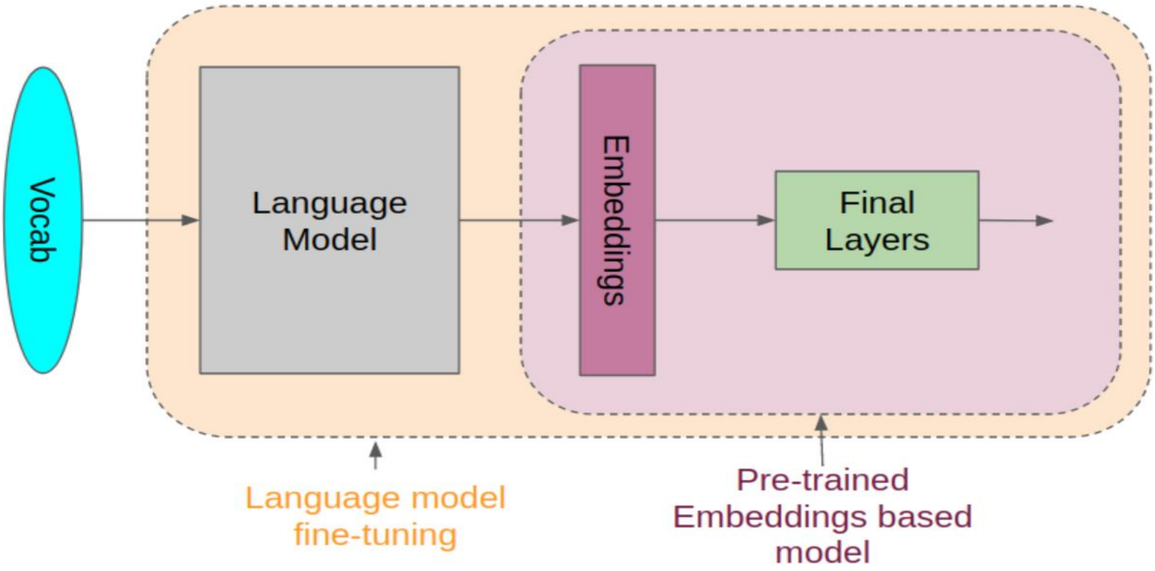
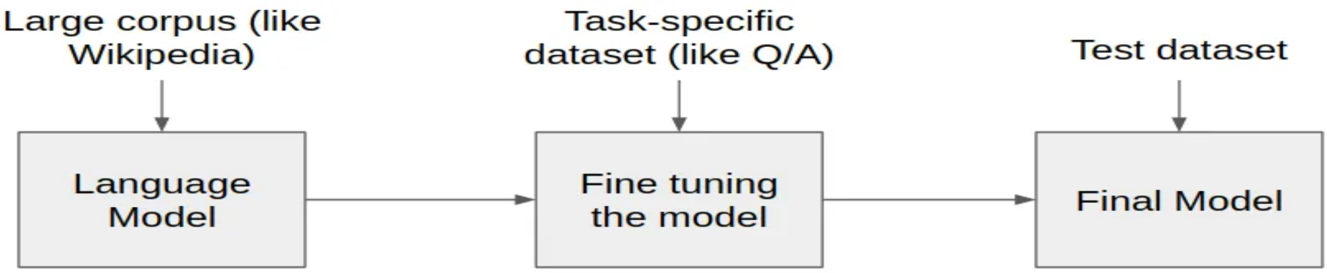

Vocab
Language Model
Embeddings
Final Layers
Language model fine-tuning
Pre-trained Embeddings based model

Fig 9: Pre-trained Models

The intuition behind pre-trained language models is to create a black box which understands the language and can then be asked to do any specific task in that language. The idea is to create the machine equivalent of a 'well-read' human being.


Large corpus (like Wikipedia)
Task-specific dataset (like Q/A)
Test dataset
Language Model
Fine tuning the model
Final Model

The language model is first fed a large amount of unannotated data (for example, the complete Wikipedia dump). This lets the model learn the usage of various words and how the language is written in general. The model is now transferred to an NLP task where it is fed another smaller task-specific dataset, which is used to fine tune and create the final model capable of performing the aforementioned task.

Fine-tuning and Transfer Learning Techniques:

The fine-tuning technique is used to optimize a model's performance on a new or different task. It is used to tailor a model to meet a specific need or domain, say cancer detection, in the field of healthcare. Pre-trained models are fine-tuned by training them on large amounts of labeled data for a certain task, such as Natural Language Processing (NLP) or image classification. Once trained, the model can be applied to similar new tasks or datasets with limited labeled data by fine-tuning the pre-trained model.

The fine-tuning process is commonly used in transfer learning, where a pre-trained model is used as a starting point to train a new model for a contrasting but related task. A pre-trained model can significantly diminish the labeled data required to train a new model, making it an effective tool for tasks where labeled data is scarce or expensive.

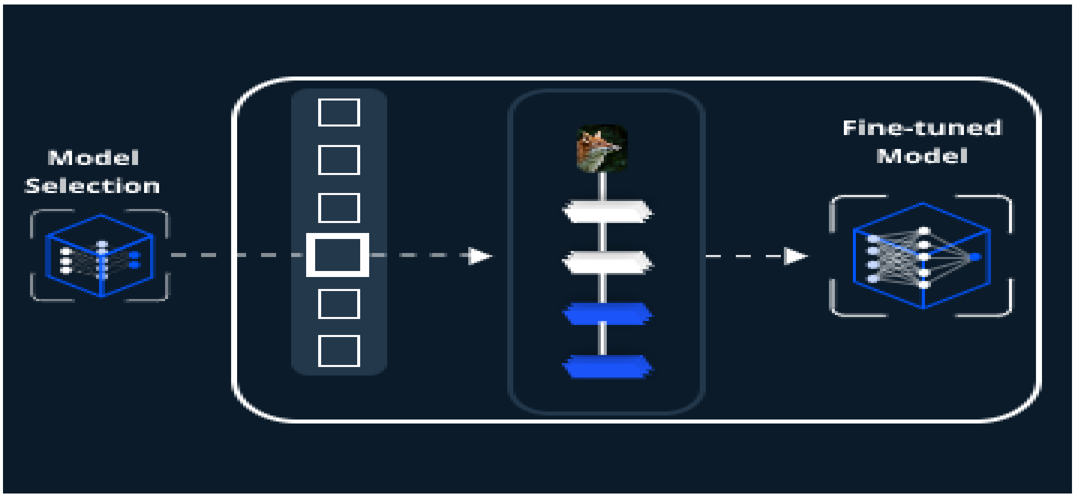


Fig 10: Fine-tuned Model

Fine-tuning a pre-trained model works by updating the parameters utilizing the available labeled data instead of starting the training process from the ground up. The following are the generic steps involved in fine-tuning:

Loading the pre-trained model:

The initial phase in the process is to select and load the right model, which has already been trained on a large amount of data, for a related task.

Modifying the model for the new task:

Once a pre-trained model is loaded, its top layers must be replaced or retrained to customize it for the new task. Adapting the pre-trained model to new data is necessary because the top layers are often task specific.

Freezing particular layers:

The earlier layers facilitating low-level feature extraction are usually frozen in a pre-trained model. Since these layers have already learned general features that are useful for various tasks, freezing them may allow the model to preserve these features, avoiding overfitting the limited labeled data available in the new task.

Training the new layers:

With the labeled data available for the new task, the newly created layers are then trained, all the while keeping the weights of the earlier layers constant. As a result, the model's parameters can be adapted to the new task, and its feature representations can be refined.

Fine-tuning the model:

Once the new layers are trained, you can fine-tune the entire model on the new task using the available limited data.

Applications of Transfer Learning in NLP:

Transfer learning has become an essential technique in the artificial intelligence (AI) domain due to the emergence of deep learning and the availability of large-scale datasets.

This comprehensive guide will discuss the fundamentals of transfer learning, explore its various types, and provide step-by-step instructions for implementing it. We'll also address the challenges and practical applications of transfer learning.

Standard NLP tools which are traditionally leveraged to train language models lack the capability of coping with novel text forms such as social media messages. Other text forms such as product reviews have different combinations of words and phrases to express the same opinion but in varying contexts.

Transfer learning helps recognize the nuances in various text forms by pretraining models to adapt to new representations of labeled data. Even in the case of unlabeled data, transfer learning provides the same accuracy in terms of sentiment recognition. Transfer learning is being used to address the challenges faced by the following areas of research:

Named Entity Recognition:

Entities are the most important components of a sentence which include nouns, verbs, noun phrases, verb phrases, or all of these. Named Entity Recognition (NER) is an NLP technique that is pre-trained to scan entire articles, textual web page content on social media, and product or service reviews to pull out fundamental entities. NER can help answer various questions such as which organizations were mentioned in an article, products mentioned in reviews, and names of persons and their locations mentioned in social media posts.

These entities are then classified into predefined categories, e.g., product names, organizations, dates, times, quantities, and amounts, and then, consequently, stored in databases. Automated chatbots and content analyser's are two of the most well-recognized applications of NER. Transfer learning uses parameter transfer methods for NER, effectively eliminating the need for lengthy exponential dataset searches.

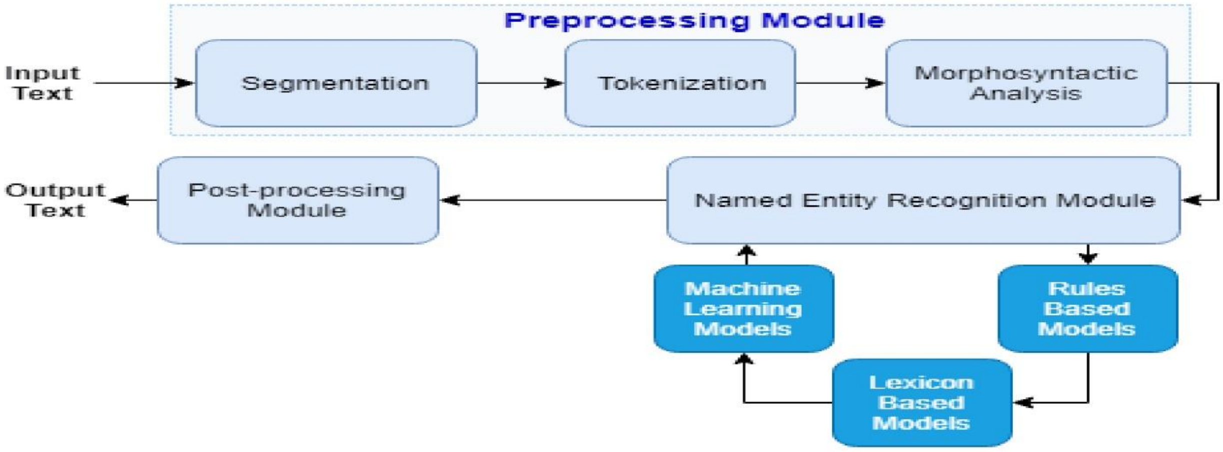


Fig 11: Preprocessing Module

Intent Classification:

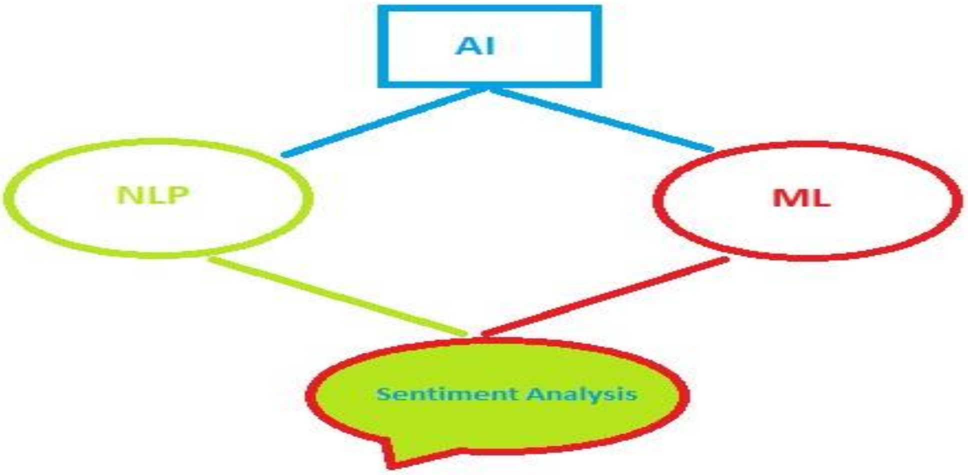
Some types of tasks and domains involve very low training data availability for target classes. This is most often true for practical sequence classification tasks in NLP. Tasks such as language modeling are used to pre-train embeddings involved in transfer learning. Under a meta-learning paradigm, transfer learning is applied to a series of related tasks using prototypical networks.

The performance variable of classification linked with intent classification increases manifold by introducing transfer learning methods. Sampling bias is also reduced by combining transfer learning-based data augmentation with meta-learning. Meta-learning means learning to learn, which is a learning paradigm within transfer learning which leverages common knowledge among a range of tasks.

Sentiment Analysis:

Sentiment analysis is a type of NLP-based textual analysis that quantifies the emotional state or the subjective information within a piece of text. Transfer learning allows for sentiment analysis with augmented data as well as little to no labeling having been done for the data.

Just like transformers aim to solve sequence-to-sequence tasks, transfer learning takes the sentiment analytics report generated from one task and extends it to another.



Similar results can be generated by fine-tuning transformers to the point that non-labeled data will get the job done too. In the case of labeled data, it can be encoded and then spit into training, testing, and validation components.

Cross-Lingual Learning:

Identifying user intents for any AI-powered analysis framework is never limited to only one language. However, the AI's understanding of the intent of user reviews or opinions from one linguistic background is often hard to duplicate with another language or even dialect. Even a slight difference in dialectic nuances completely throws off the AI's capability to understand the subjective meaning of textual data.

Computational linguists function in a domain that involves the scientific study of language from a computational outlook. It helps in day-to-day functions in AI-centered work structures such as machine translation, speech synthesis, grammar checking, text mining, and so on.

Sequence Labeling:

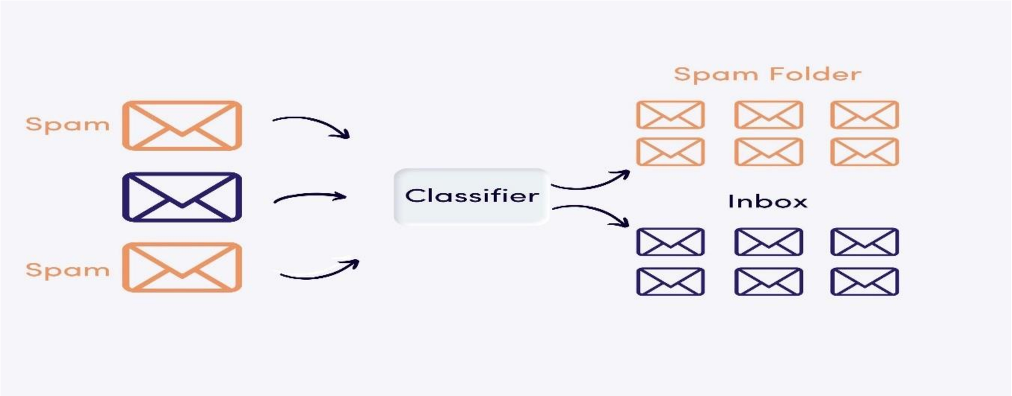
In sequence labeling, the AI models progressively learn to maximize the conditional probability of certain outcomes in textual analysis. A standard sequence labeling problem takes a given input, i.e., a text form, and predicts the output sequence that produces named entities. Based on this, the various entities in an article, review, or social media post are categorized into their specific descriptive fields.

A word in the input is characterized by both its word-level and character-level representations while being fed into a transfer learning AI model. A lookup table trained with transfer learning is then employed to map the input words in an exponential search. The entire sequence labeling workflow is shortened drastically by the application of transfer learning.

Real-world examples and case studies:

Email filters are common NLP examples you can find online across most servers.

Spam filters are where it all started they uncovered patterns of words or phrases that were linked to spam messages. Since then, filters have been continuously upgraded to cover more use cases.



An NLP case study we can look at is gmail’s new classification system. This upgraded system categorizes emails into one of three groups (primary, social, or promotions) based on the email content. This is a convenient application of NLP that keeps Gmail users' inboxes under control while highlighting relevant and high-priority emails.

Levity offers its own version of email Classification through using NLP. This way, you can set up custom tags for your inbox and every incoming email that meets the set requirements will be sent through the correct route depending on its content.

Challenges and Limitations:

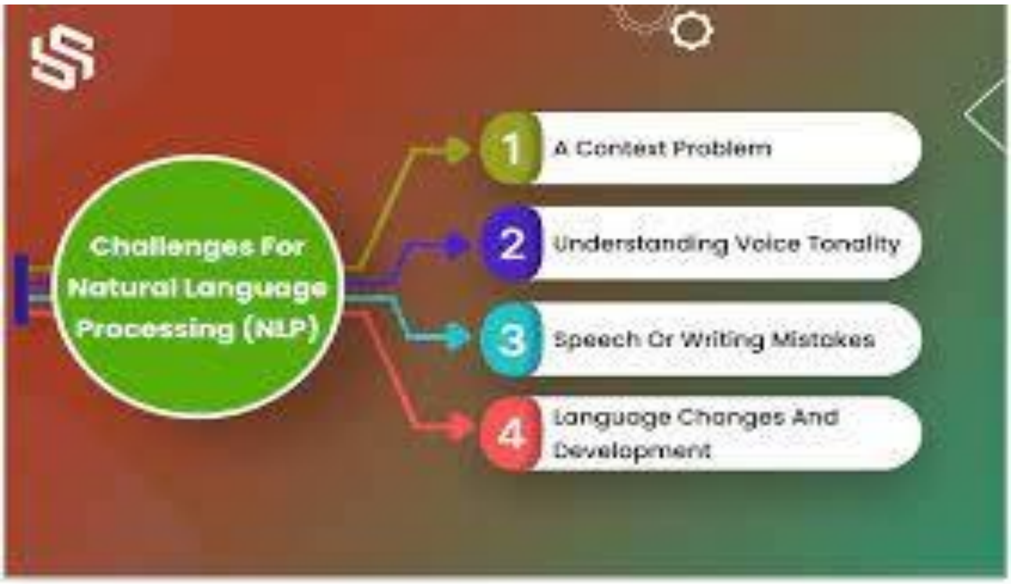
Invaluable support for artificial intelligence (AI), natural language processing (NLP) helps in establishing effective communication between computers and human beings. In recent years, there have been significant breakthroughs in empowering computers to understand human language using NLP. However, the complex diversity and dimensionality characteristics of the data sets, make this simple implementation a challenge in some cases.

Challenges for NLP implementation:

Data challenges:

The main challenge is information overload, which poses a big problem to access a specific, important piece of information from vast datasets. Semantic and context understanding is essential as well as challenging for summarisation systems due to quality and usability issues. Also, identifying the context of interaction among entities and objects is a crucial task, especially with high dimensional, heterogeneous, complex and poor-quality data.

Data ambiguities add more challenges to contextual understanding. Semantics are important to find the relationship among entities and objects. Entities and object extraction from text and visual data could not provide accurate information unless the context and semantics of interaction are identified. Also, the currently available search engines can search for things (objects or entities) rather than keyword-based search. Semantic search engines are needed because they better understand user queries usually written in natural language.


Challenges For Natural Language Processing (NLP)
1 A Context Problem
2 Understanding Voice Tonality
3 Speech Or Writing Mistakes
4 Language Changes And Development

Text related challenges:

Large repositories of textual data are generated from diverse sources such as text steams on the web, communications through mobile and IoT devices. Though ML and NLP have emerged as the most potent and most used technology applied to the analysis of the text and text classification remains the most popular and the most used technique. Text classification could be Multi-Level (MLC) or Multi-Class (MCC). In MCC, every instance could be assigned to only one class label, whereas MLC is a classification that assigns multiple labels to a single instance.

Solving MLC problems requires an understanding of multi-label data pre-processing for big data analysis. MLC can become very complicated due to the characteristics of real-world data such as high-dimensional label space, label dependency, and uncertainty, drifting, incomplete and imbalanced. Data reduction for large dimensional datasets and classifying multi-instance data is also a challenging task.

Then there are the issues posed by a language translation. The main challenge with language translation is not in translating words, but in understanding the meaning of sentences to provide an accurate translation. Each text comes with different words and requires specific language skills. Choosing the right words depending on the context and the purpose of the content, is more complicated.

Future Directions:

Transfer learning is a rapidly evolving field, and there are a number of potential future developments in transfer learning for NLP. Some of these developments include:

- Improved domain adaptation techniques: Domain adaptation techniques are already being used to improve the performance of transfer learning models on new domains. However, there is still room for improvement in this area. Researchers are developing new domain adaptation techniques that are more robust to domain shift and that can be applied to a wider range of NLP tasks.
- More efficient fine-tuning techniques: Fine-tuning pre-trained models can be computationally expensive, especially for large models. Researchers are developing new fine-tuning techniques that are more efficient and that can be used to fine-tune pre-trained models on smaller datasets.
- New transfer learning techniques for low-resource languages: Transfer learning has been shown to be effective for improving the performance of NLP models on low-resource languages. However, there is still room for improvement in this area. Researchers are developing new transfer learning techniques that are specifically designed for low-resource languages.

In addition to these general developments, there are a number of emerging techniques and research areas in transfer learning for NLP, such as:

- Zero-shot learning: Zero-shot learning is a type of transfer learning where the model is able to perform a task without being trained on any labeled data for that task. This is achieved by transferring knowledge from a related task where the model has been trained on labeled data.
- Few-shot learning: Few-shot learning is a type of transfer learning where the model is able to perform a task after being trained on a very small number of labeled data for that task. This is achieved by

leveraging the knowledge that the model has learned from a related task where it has been trained on a large dataset.

These emerging techniques have the potential to revolutionize the way that we develop and deploy NLP models. For example, zero-shot learning could be used to develop NLP models for new languages or new tasks without the need to collect and label large datasets of data. Few-shot learning could be used to develop NLP models that can be quickly adapted to new tasks or new domains.

Conclusion:

Most of the challenges are due to data complexity, characteristics such as sparsity, diversity, dimensionality, etc. and the dynamic nature of the datasets. NLP is still an emerging technology, and there are a vast scope and opportunities for engineers and industries to deal with many open challenges of implementing NLP systems.

With the special focus on addressing NLP challenges, organisations can build accelerators, robust, scalable domain-specific knowledge bases and dictionaries that bridges the gap between user vocabulary and domain nomenclature. The proficient and skilled pool of data scientists working for any product engineering services provider will be capable of building customised architectures and NLP pipeline to enable NLP search on different kind of datasets (structured & unstructured).

The quality research and adoption of state-of-the-art technologies like linked data, knowledge graph, etc., can improve the quality of data with enriched meanings, linking data sources with appropriate and meaningful relationships among them. Last but not least, developing accelerators and frameworks make complex NLP implementations more affordable and provide improved performance.

References

[1] Neil Houlsby et al. "Parameter-efficient transfer learning for NLP." Preprint, arXiv, 2019.

[2] Jeremy Howard and Sebastian Ruder. "Fine-tuned language models for text classification." Preprint 1, arXiv, 2018.

[3] Dichao Hu. "An introductory survey on attention mechanisms in NLP problems." In Proceedings of SAI Intelligent Systems Conference, pages 432–448. Springer, 2019.

[4] Michael Hüsken and Peter Stagge. "Recurrent neural networks for time series classification." Neurocomputing, 50.

[5] Oliver Adams, Adam Makarucha, Graham Neubig, Steven Bird, and Trevor Cohn. "Cross-lingual word embeddings for low-resource language modeling." In Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers, pages 937–947.

[6] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. "Neural machine translation by jointly learning to align and translate." CoRR, abs/, 1409:0473, 2014.

[7] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. "Curriculum learning." In Proceedings of the 26th annual international conference on machine learning, pages 41–48, 2009.

[8] Samuel R Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. "A large annotated corpus for learning natural language inference." arXiv preprint arXiv: 1508.05326, 2015.

[9] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. "Language models are few-shot learners," 2020.

[10] Jacob Devlin et al. "Bert: Pre-training of deep bidirectional transformers for language understanding." Preprint, arXiv, 2018.