How to Use **BERT** to Improve the Quality of **Entity Annotation in ML Models**

Location

Product

Person

hitech
Your Partner in Digital Excellence | Since 1992

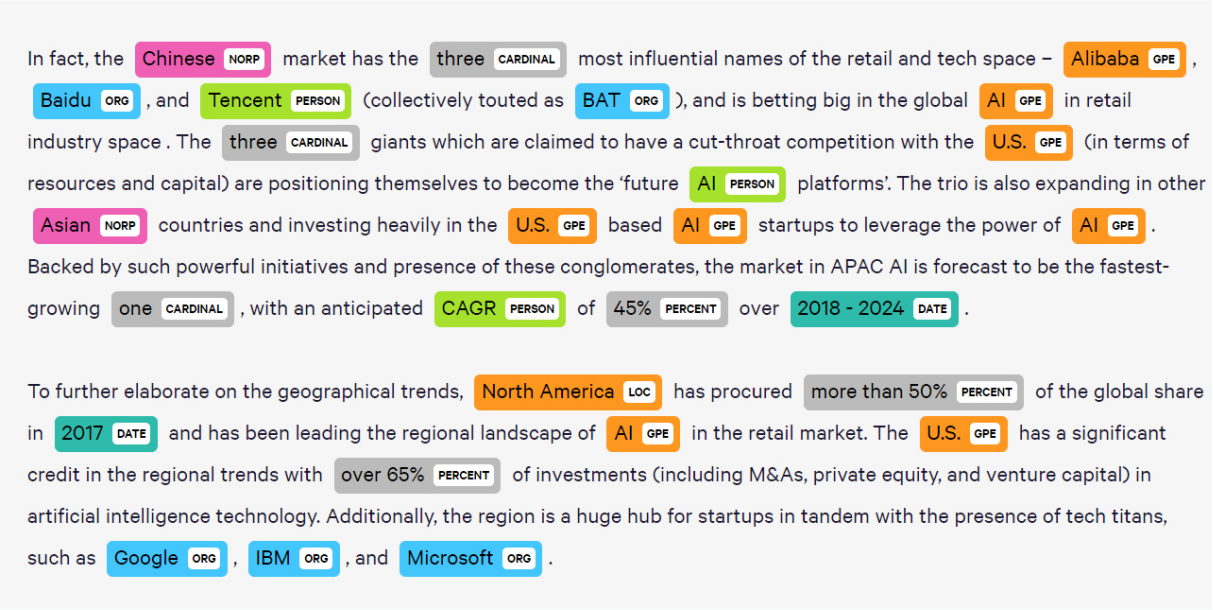
BERT significantly reduces the time required for entity annotation in NLP projects without compromising labeling quality. Aligning domain-specific BERT models with Named Entity Recognition projects is a strategic move.

Machine learning models often face challenges in real-life scenarios if they struggle to correctly identify entities. Poor entity annotation in training data is a primary reason for such failures, especially when trying to manage time and cost constraints.

AI and machine learning projects often require the identification of more than basic entities, necessitating custom entity recognition. This can lead to increased project costs and time when using older methods or CycleNER for entity annotation. However, with BERT, achieving a higher level of custom entity recognition becomes feasible with just 400-500 samples of labeled data. This makes text annotation for machine learning much more manageable in terms of quality, time, and budgets.

As the project head of entity recognition and annotation for my company, I continuously seek ways to address the lack of labeled data and improve accuracy in custom entity recognition. BERT has revolutionized our ability to build custom models even for individual entities. We can perform pre-training with large masses of unannotated text and then fine-tune models to suit project requirements, using minimal sample data. Moreover, we can now find pre-trained BERT models tailored to different fields, which we can fine-tune according to the instructions of the AI and ML companies we collaborate with.

TABLE OF CONTENTS



What is entity annotation in machine learning?

Entity annotation is the process we use to create labeled training data for ML models. It is also called entity recognition or named entity recognition - NER. It is a crucial task in machine learning, especially in the field of natural language processing (NLP). In entity annotation, we use both manual and automated processes to identify and classify entities, such as names of people, organizations, locations, dates, and much more that are within unstructured text data.

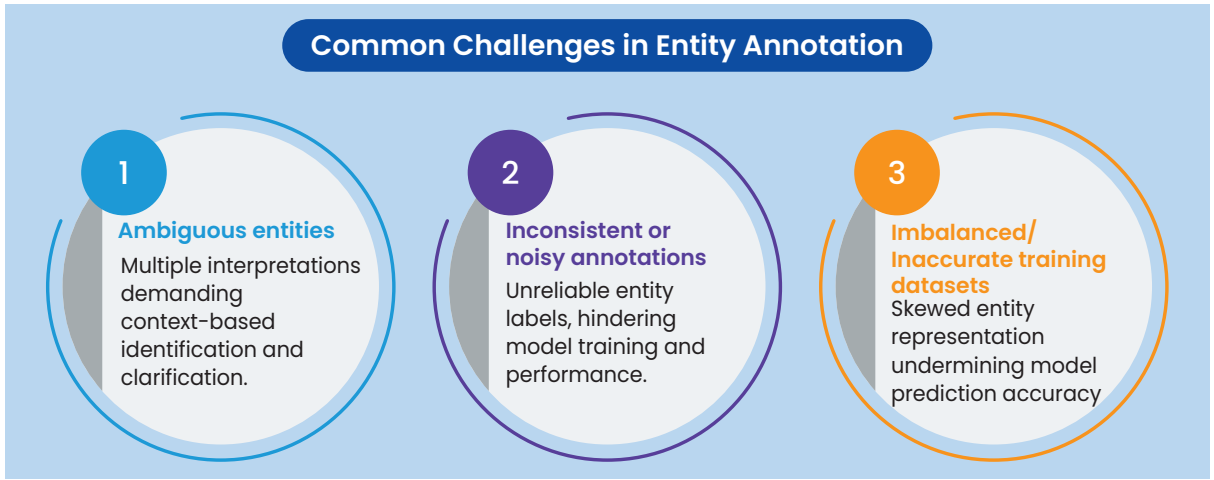
Annotation service providers handle huge volumes of text, much of which is irrelevant to the domain or language of the service provider. And often, substandard, or poorly annotated training datasets become the principal cause of unreliable ML model performance.

> So, we at Hitech BPO look at unreliable output of ML models primarily as a data annotation problem rather than an algorithmic problem. Algorithms will evolve and improve over time, but their performance relies on the high-quality training material we deliver to AI and ML companies.

As a leading data annotation service provider, we always keep in mind that machine learning techniques ultimately rely on data annotated by humans, whether it be text annotation for machine learning, entity annotation, imag e recognition, machine translation, or for that matter any NLP algorithm. And this is why we place high value on integrating human expertise in data annotation, both for initial ML training datasets and for human-in-the-loop setups for critical AI applications.

What are the common challenges in entity annotation?

We often encounter challenges in entity annotation which impact the efficiency, quality, and overall success of the ML algorithm. Here are some of them:



Ambiguous entities

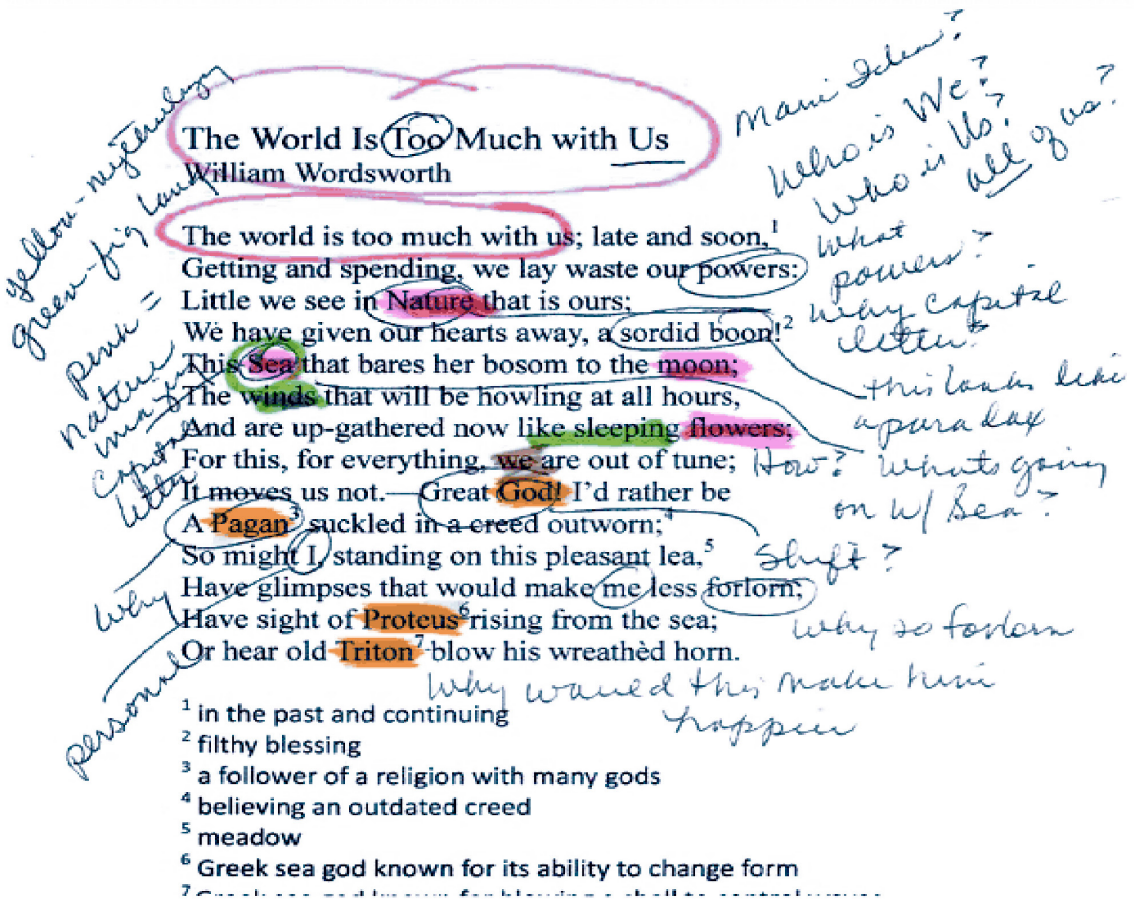
- **Polysemy:** Words or phrases with multiple meanings depending on the context they are used in. For example, the word 'apple' could refer to the fruit or the technology company or even an emotion like affection in the expression 'an apple of my eye'.
- **Homonyms:** Words that have the same spelling and pronunciation but different meanings. For example, the word 'bank' could refer to a financial institution or the side of a river.
- **Abbreviations and acronyms:** Words that represent different entities depending on the context. For instance, 'CIA could refer to the Central Intelligence Agency or the Culinary Institute of America.
- **Named entity variations:** Entities such as people or organizations have multiple variations in spelling, capitalization, or punctuation. For example, 'International Business Machines' and 'IBM' both refer to the same organization.
- **Lack of context:** Words or phrases that don't provide enough information to disambiguate it accurately leads to incorrect annotations or the need for additional information to make an accurate decision.
- **Language nuances:** Idiomatic expressions, slang, or cultural references act as barriers in accurately identifying and handling ambiguous entities.

Inconsistent or noisy annotations

- **Subjectivity:** Human annotators can interpret guidelines differently due to varying levels of expertise and cultural conditioning. This leads to subjective decisions and inconsistencies in the annotated data.
- **Ambiguity:** Inconsistencies arise when different annotators interpret the context differently, or when the context itself is insufficient to disambiguate the entity.
- **Annotation guidelines:** Annotators not equipped with precise instructions and examples struggle to make consistent decisions, leading to noisy annotations.
- **Annotator fatigue:** Annotators get tired or lose focus which makes the quality of their annotations decline, resulting in inconsistencies and errors.
- **Language and domain expertise:** Annotators with varying levels of language proficiency or domain expertise, or annotators who are not familiar with specific domain terminology or language nuances struggle to identify and classify entities accurately.

Imbalanced and inaccurate training datasets

- **Imbalanced data:** Certain entity classes may be underrepresented in the training data. This imbalance can lead to biased machine learning models that perform poorly on underrepresented classes, resulting in lower overall accuracy and reduced effectiveness in real-world applications.
- **Inaccurate annotations:** Human errors, ambiguous entities, or unclear annotation guidelines lead to inaccurate annotations resulting in incorrect patterns or relationships between entities.



How do entity annotation challenges impact the performance of ML models?

Poor entity annotation can negatively impact model performance, leading to issues such as poor generalization, overfitting, underfitting, unfairness, and misinterpretation of relationships between features and target variables.

Some of the main consequences of poor annotation on ML performance include:

- **Poor generalization:** Learning incorrect patterns or relationships results in poor generalization of new, unseen data. It leads to reduced accuracy and unreliability of the ML model.
- **Overfitting:** Errors or noise in the training data handicap the ML models in generalization. They become incapable of learning the underlying patterns.
- **Underfitting:** Insufficient training data makes the ML model incapable of capturing the underlying patterns in the training data, and the model becomes less able (underfit) at making accurate predictions.
- **Unfairness:** Models trained on a dataset with gender or racial biases, will produce biased predictions that perpetuate existing inequalities and discrimination.
- **Misinterpretation:** Erroneous training data causes ML models to misinterpret the relationships between features and the target variable. Consequences can be serious in applications such as medical diagnosis, financial decision-making, or autonomous systems.

> We tackle most of these challenges by using a combination of human expertise and machine learning algorithms. We also incorporate external knowledge sources like knowledge graphs and domain-specific databases to disambiguate entities and improve the overall quality of entity annotation.

It's important to note that while machine learning models can learn from annotated data to improve their ability to handle ambiguity; human annotators often exhibit limitations when it comes to context, language, and domain knowledge to disambiguate entities.

Using BERT-based models has helped us achieve higher accuracy and better generalization in entity recognition tasks for ML models.

Want to overcome entity annotation challenges?

Consult us now »

How can BERT overcome the challenges in entity annotation?

BERT tackles entity annotation challenges by contextualizing entities in text, capturing word relationships, and disambiguating meanings. Its bidirectional nature enables better understanding of complex entity references. By fine-tuning annotated data, BERT adapts to specific domains and improves entity annotation. Let's look at BERT, its usage and how it helps in addressing entity annotation challenges.

What is BERT and how does it work?

BERT (Bidirectional Encoder Representations from Transformers) is a state-of-the-art NLP model that captures deep contextual information from text. Pre-trained on a large corpus of text, BERT can be fine-tuned for various NLP tasks, including entity annotation. It addresses challenges by offering bidirectional context, transfer learning, attention mechanisms, and multi-task learning capabilities.

> In their **BERT paper,** authors Jacob Devlin, Ming-Wei Chang, Kenton Lee, & Kristina Toutanova suggest that "BERT is conceptually simple & empirically powerful. It obtains new state-of-the-art results on natural language processing tasks with 7.7% absolute improvement and 86.7% accuracy.

BERT has helped Hitech BPO fast track our natural language processing - NLP projects. And by leveraging BERT's bidirectional context, pre-training, transfer learning, attention mechanism, and multi-task learning capabilities, we can address the challenges associated with regular entity annotation.

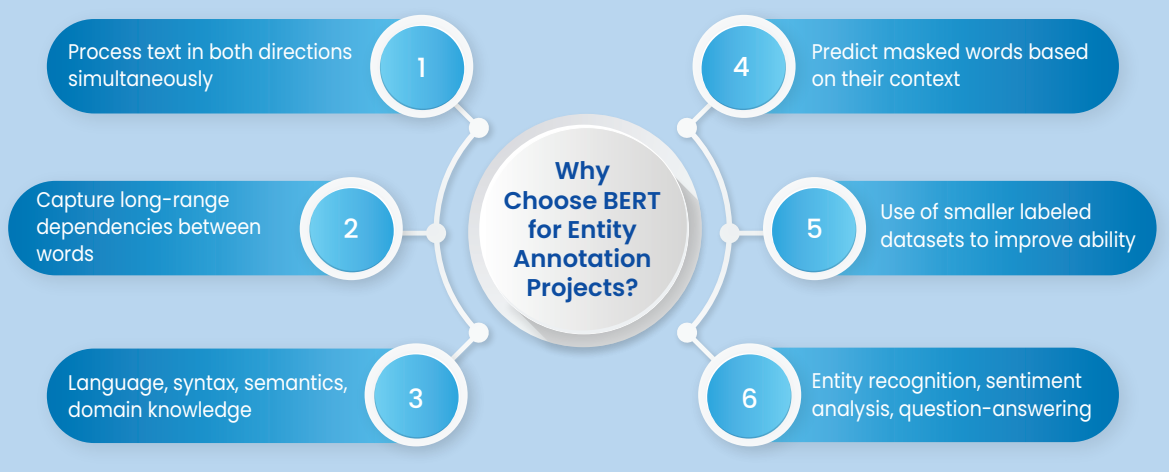
Why is BERT better for NER (Named Entity Recognition)?

BERT is better than earlier methods, as unsupervised entity annotation using CycleNER has always been a challenge. CycleNER cannot process unannotated sentences without a small amount of accurate named entity recognition sample training data.

BERT is already pre-trained on large amounts of unannotated data, and it successfully covers most entity annotation needs.

Once machine learning models are trained with accurately annotated samples of labeled data created using BERT, they can then automatically recognize and classify entities in new text. This process is essential for NLP tasks and applications such as information extraction, sentiment analysis, question-answering systems, text summarization, machine translation, and many more.

Key features of BERT that make it a first choice for entity annotation projects



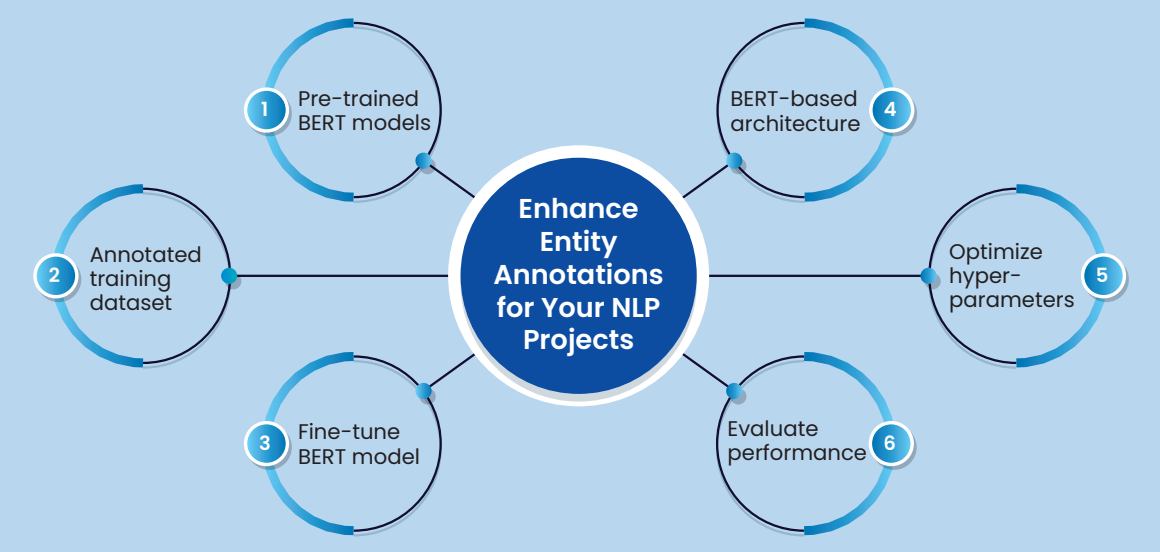
- **Bidirectional context:** Unlike traditional language models that process text in a single direction (either left-to-right or right-to-left), BERT is designed to capture bidirectional context by processing text in both directions simultaneously.
- **Transformer architecture:** It employs self-attention mechanisms to weigh the importance of different words in context when making predictions, enabling it to focus on the most relevant information for disambiguating entities and capture long-range dependencies between words.
- **Pre-training on large-scale data:** It comes with rich representation of language, syntax, semantics, and domain-specific knowledge. This helps BERT generalize better to new tasks and domains making it more effective at capturing contextual information in entity annotation tasks.
- **Masked Language Modeling (MLM):** In this phase it learns to predict randomly masked words in a sentence based on their context. This compels the BERT model to learn bidirectional context and capture meaningful relationships between words, which is crucial for understanding the context in entity annotation tasks.

> The key feature of BERT is that a deep bidirectional model can be pre-trained using only a plain text corpus (i.e., without any annotations), and then fine-tuned on a relatively small amount of labeled data to achieve state-of-the-art performance on a wide range of tasks. Says Jay Alammar, a **machine learning researcher.**

- **Transfer learning:** The approach of fine-tuning the pre-trained model on a specific task using a smaller labeled dataset allows BERT to benefit from the knowledge gained during pre-training, reducing the amount of labeled data required for fine-tuning and improving its ability.
- **Multi-task learning:** Its capability to be fine-tuned for multiple NLP tasks such as entity recognition, sentiment analysis, and question-answering simultaneously is commendable. It empowers the BERT model to learn shared representations across tasks, improve performance on individual tasks, and effectively capture contextual information.

How can you use BERT to improve entity annotation?

Here's a step-by-step guide on how to use BERT for enhancing entity annotation:



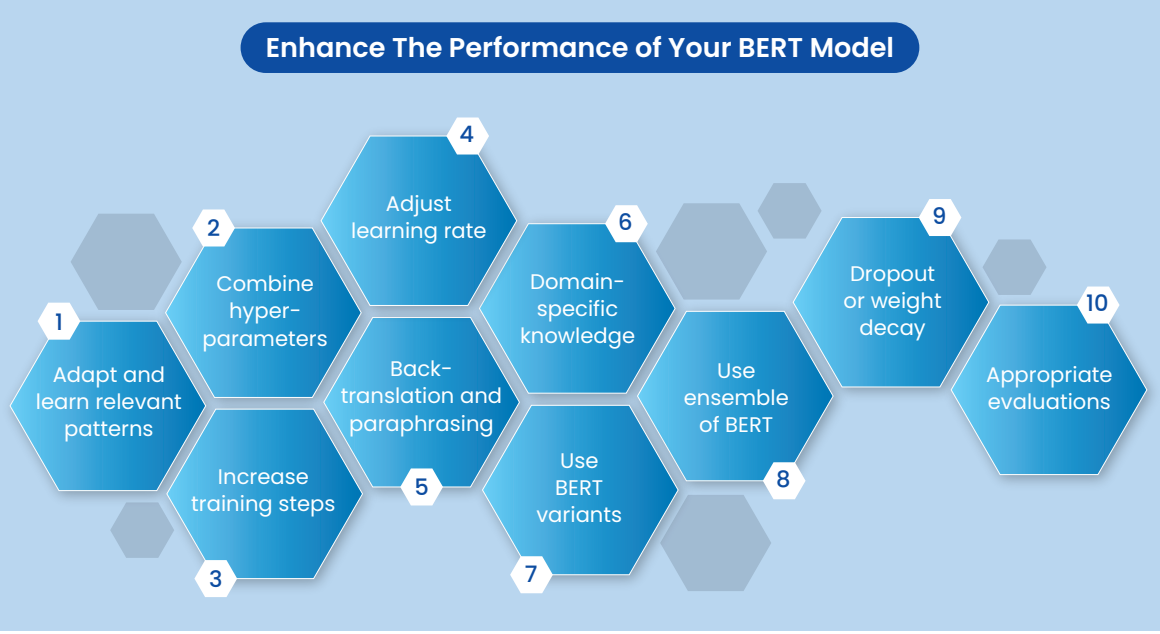
- **Leverage pre-trained BERT models:** Start by using a pre-trained BERT model, such as BERT-base or BERT-large, which has already been trained on large-scale text corpora like Wikipedia and BooksCorpus. These pre-trained models have learned rich contextual representations that can be fine-tuned for specific entity annotation tasks.
- **Prepare annotated data:** Collect and annotate entities in text data relevant to your domain or task. Ensure that the annotations are of high quality and consistent to avoid introducing errors or biases into the model.
- **Fine-tune BERT on your dataset:** Fine-tune the pre-trained BERT model on your annotated dataset. During fine-tuning, the model will learn to adapt its contextual understanding to the specific entities and relationships present in your data. This process involves updating the model's weights using backpropagation and gradient descent based on the annotated examples.
- **Experiment with different BERT-based architectures:** Depending on the complexity of your entity annotation task, you may want to experiment with different BERT-based architectures, such as BERT-CRF (Conditional Random Field), BERT-BiLSTM (Bidirectional Long Short-Term Memory), or BERT-MLP (Multilayer Perceptron). These architectures combine BERT's contextual representations with additional layers or techniques to improve sequence modeling and entity classification.
- **Optimize hyperparameters and model architecture:** To further improve your BERT model's performance, experiment with different hyperparameters, such as learning rate, batch size, and the number of training epochs. Additionally, you can adjust the model architecture, such as adding or modifying layers, to better suit your specific entity annotation task.
- **Evaluate and iterate:** After fine-tuning your BERT model, evaluate its performance on a separate validation dataset to ensure it generalizes well to unseen data. Measure metrics like precision, recall, F1 score, and accuracy to assess the quality of entity annotations. Iterate on the model training and fine-tuning process to achieve the desired level of performance.

Want to reduce entity annotation time in your NLP projects?

Consult our experts »

Strategies for improving BERT models

To enhance the performance of your bert modelTo enhance BERT models, consider strategies like fine-tuning, hyperparameter optimization, longer training, learning rate scheduling, data augmentation, domain adaptation, model architecture exploration, ensemble methods, and regularization. Proper evaluation metrics should be used to assess the model's performance accurately.



Fine-tuning: Ensure that you are fine-tuning the BERT model on a task-specific labeled dataset. The fine-tuning process allows the pre-trained BERT model to adapt to the specific task and learn relevant patterns from the labeled data.

Hyperparameter optimization: You can experiment with different hyperparameters during fine-tuning, such as learning rate, batch size, number of epochs, and weight decay. You can also conduct a systematic search (e.g., grid search or random search) to find the optimal combination of hyperparameters that yields the best performance.

Longer training: This is what we practice at Hitech BPO. We increase the number of training epochs or steps to allow the model more time to learn from the data. However, we always keep a tab on, and monitor the model's performance on a validation set to determine the optimal stopping point and avoid overfitting.

Learning rate scheduling: Using learning rate scheduling strategies like linear warm-up and cosine annealing in order to adjust the learning rate during training is also a very good idea. It will help the model to converge faster and achieve better performance.

Data augmentation: We apply data augmentation techniques to increase the size and diversity of our training datasets, and we recommend you also to do so. For text classification tasks, this may involve techniques such as synonym replacement, back-translation, or paraphrasing. Data augmentation can help your model generalize better and improve its performance on unseen data.

Domain adaptation: If your task involves a specific domain, consider using a BERT model that has been pre-trained on domain-specific data (e.g., BioBERT for biomedical text or SciBERT for scientific text). Domain-adapted BERT models can capture domain-specific knowledge and provide better performance for tasks within that domain.

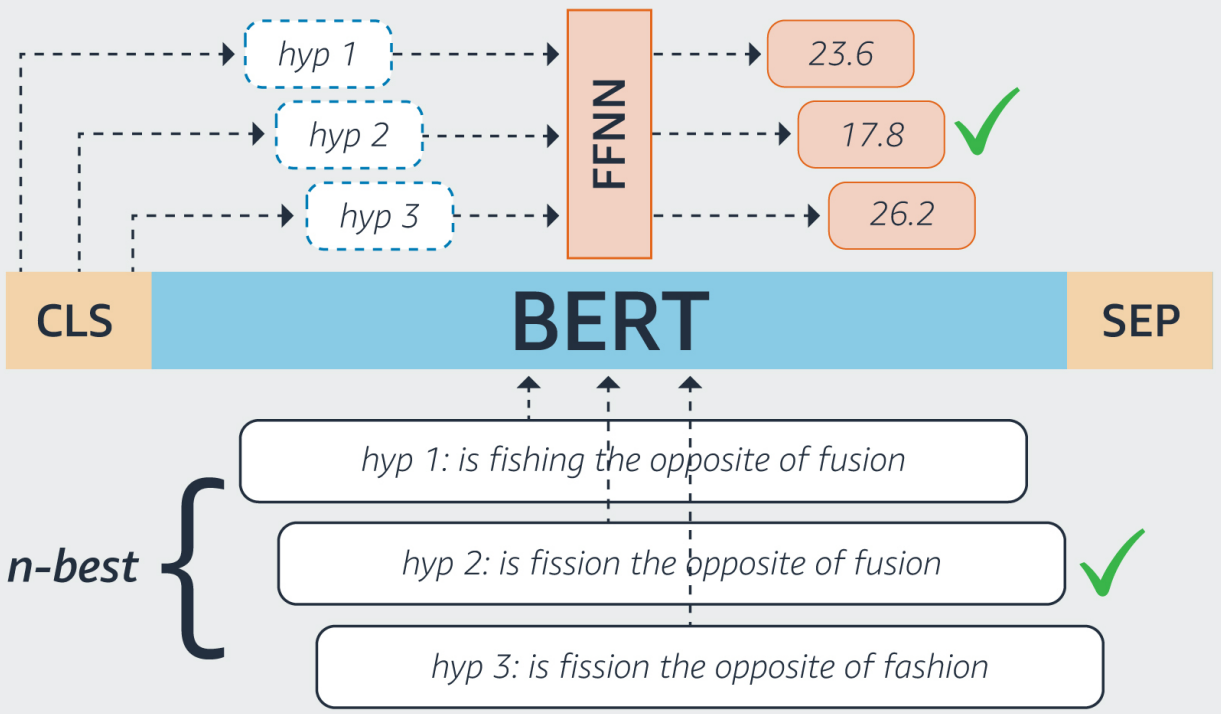
Model architecture: Experiment with different BERT variants, such as BERT-base, BERT-large, or more recent models like RoBERTa, ALBERT, or DeBERTa. These variants may offer architectural improvements or optimizations that can lead to better performance for your specific task.

Ensemble methods: Combine the predictions of multiple BERT models or use an ensemble of BERT models with other types of models (e.g., LSTM, CNN) to improve overall performance. Ensemble methods can leverage the strengths of different models to achieve better results.

Regularization: Apply regularization techniques, such as dropout or weight decay, to prevent overfitting and improve the model's generalization capabilities.

Evaluation metrics: Ensure that you are using appropriate evaluation metrics for your task. Some tasks may require specific metrics (e.g., F1-score for imbalanced datasets) to accurately assess the model's performance.

By experimenting with these strategies and monitoring the model's performance on a validation set, you can iteratively improve your BERT model and achieve better results for your specific task.



Which BERT model is best for entity annotation?

The choice of the best BERT model depends on factors like the domain, language, and available resources. Consider options like BERT-base, BERT-large, domain-specific BERT models, language-specific BERT models, and optimized BERT variants. Each variant has its strengths and applications.

Here are some BERT variants and specialized models that are used for case-specific entity annotation tasks:

BERT-base and BERT-large: These are the original BERT models, with BERT-base having 12 layers and 110 million parameters, while BERT-large has 24 layers and 340 million parameters. BERT-large typically provides better performance but requires more computational resources.

Domain-specific BERT models: If your entity annotation task involves a specific domain, consider using a BERT model pre-trained on domain-specific data, such as BioBERT for biomedical text or SciBERT for scientific text. These models can capture domain-specific knowledge and provide better performance for tasks within that domain.

Language-specific BERT models: For non-English entity annotation tasks, consider using language-specific BERT models, such as multilingual BERT (mBERT) or models pre-trained on specific languages (e.g., BERTje for Dutch, CamemBERT for French). These models are designed to capture the nuances of different languages and can provide better performance for entity annotation tasks in those languages.

> Domain-specific pre-trained models like BioBERT and SciBERT have been shown to outperform the original BERT model on entity annotation tasks in their respective domains, demonstrating the value of domain-specific pre-training, **say authors of BioBERT paper.**

Optimized BERT variants: More recent BERT variants, such as RoBERTa, ALBERT, or DeBERTa, offer architectural improvements or optimizations that can lead to better performance. RoBERTa, for example, uses a larger training dataset and modifies the pre-training process, resulting in improved performance compared to the original BERT.

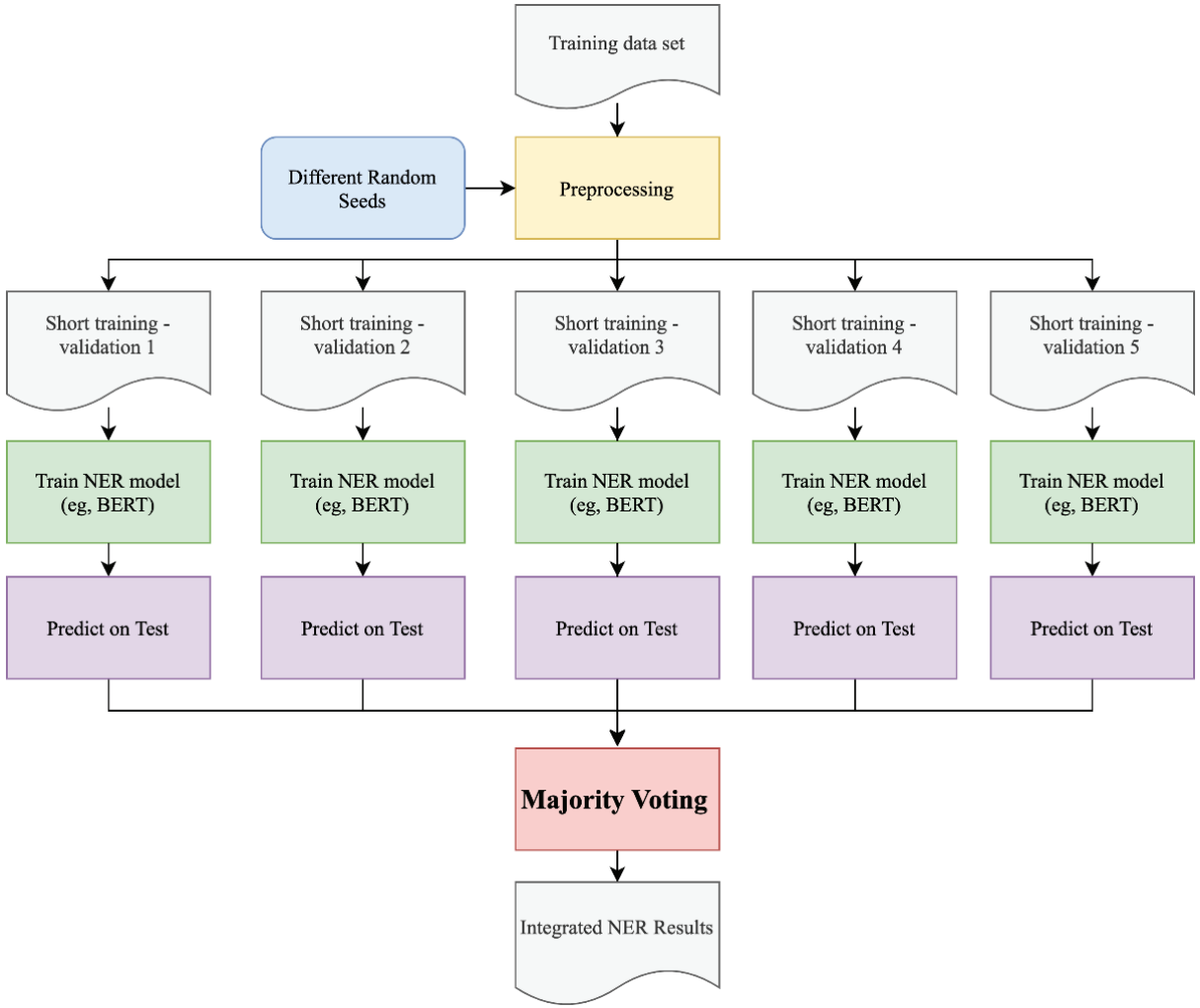
Computationally efficient models: If computational resources are a concern, consider using more efficient BERT variants, such as DistilBERT or TinyBERT. These models are smaller and faster while still providing competitive performance compared to the original BERT models.

BERT-BiLSTM models: It is a fine blend of BERT (Bidirectional Encoder Representations from Transformers) and BiLSTM (Bidirectional Long Short-Term Memory). Here the BERT component is used to generate contextualized embeddings for the input text, while the BiLSTM component processes these embeddings to capture long-range dependencies and model the sequential nature of the text.

BERT-CRF models: If tackling sequence labeling tasks, such as Named Entity Recognition (NER) and Part-of-Speech (POS) tagging are a challenge, using this combination of BERT (Bidirectional Encoder Representations from Transformers) and CRF (Conditional Random Field) is the best option. The BERT component generates contextualized embeddings for the input text, while the CRF component models the dependencies between output labels in the sequence, capturing the label transition probabilities and enforcing label consistency.

BERT-MLP models: In a BERT-MLP model, the BERT component is used to generate contextualized embeddings for the input text, while the MLP component serves as a task-specific classifier or regressor that processes these embeddings to make predictions.

To determine the best BERT model for your entity annotation task, consider the specific requirements of your task, such as domain, language, and available resources. Experiment with different BERT models and monitor their performance on a validation set to identify the model that provides the best results for your specific task.



Successful applications of BERT in entity annotation

BERT has been successfully used for sentiment analysis on Twitter, emotion categorization, clinical notes analysis, speech-to-text translation, and toxic comment detection, among others.

- **Twitter sentiment analysis:** Pre-trained BERT models are used to analyze the sentiment of tweets, helping to classify them as positive, negative, or neutral.
- **Emotion categorization:** BERT models are leveraged to categorize emotions like anger, fear, joy, etc. in English text.
- **Clinical notes analysis:** BERT has succeeded in extraction of relevant information and entities from medical records and analyzing clinical notes.
- **Speech to text translation:** BERT models have been used to enhance the performance of speech-to-text translation systems, improving the recognition of entities in transcribed text.
- **Toxic comment detection:** BERT has been utilized to detect toxic comments in online platforms, helping to identify and filter out harmful content.

Case study

An entity annotation project at Hitech BPO

While we cannot share confidential details, we can give you an example overview of an entity annotation project we did. It was focused on extracting named entities from scientific research papers for a large pharmaceutical company. We were required to annotate data for building entity models using BERT to identify and classify entities such as disease names, drug names, protein, or gene names, and so forth, which were to be used to build a knowledge graph for the company's research division.

Challenges included managing long sentences beyond BERT's token limit and the project's success led to improved efficiency in drug discovery and an evolving knowledge graph.

Context and case needs

Given the complexity and length of sentences in scientific papers, one of our main challenges was managing long sentences exceeding BERT's token limit. Often, crucial entity relationships were found in sentences that ran far beyond the 512-token limit, hence the need to ensure a comprehensive and accurate annotation.

Strategy and execution

Our approach was three-fold:

- **Tokenization and sequence splitting:** We first tokenized the sentences using BERT's WordPiece tokenizer. To tackle sentences beyond the 512-token limit, we implemented the sliding window approach, with each window containing 512 tokens and an overlap of 128 tokens. This method ensured the contextual integrity of the information we wanted to capture.
- **Annotation adjustment:** In the process of splitting, if an entity was cut off in the middle, we made the tough call of discarding such entities to ensure the quality and consistency of our training data. We believe maintaining the integrity of entities was crucial for the model to learn accurately.
- **Model training and prediction:** We prepared the model inputs, trained our BERT model, and adjusted predictions for overlapped tokens by taking the majority vote. This way, we made sure that each token's entity label was decided upon the highest consensus from multiple windows.

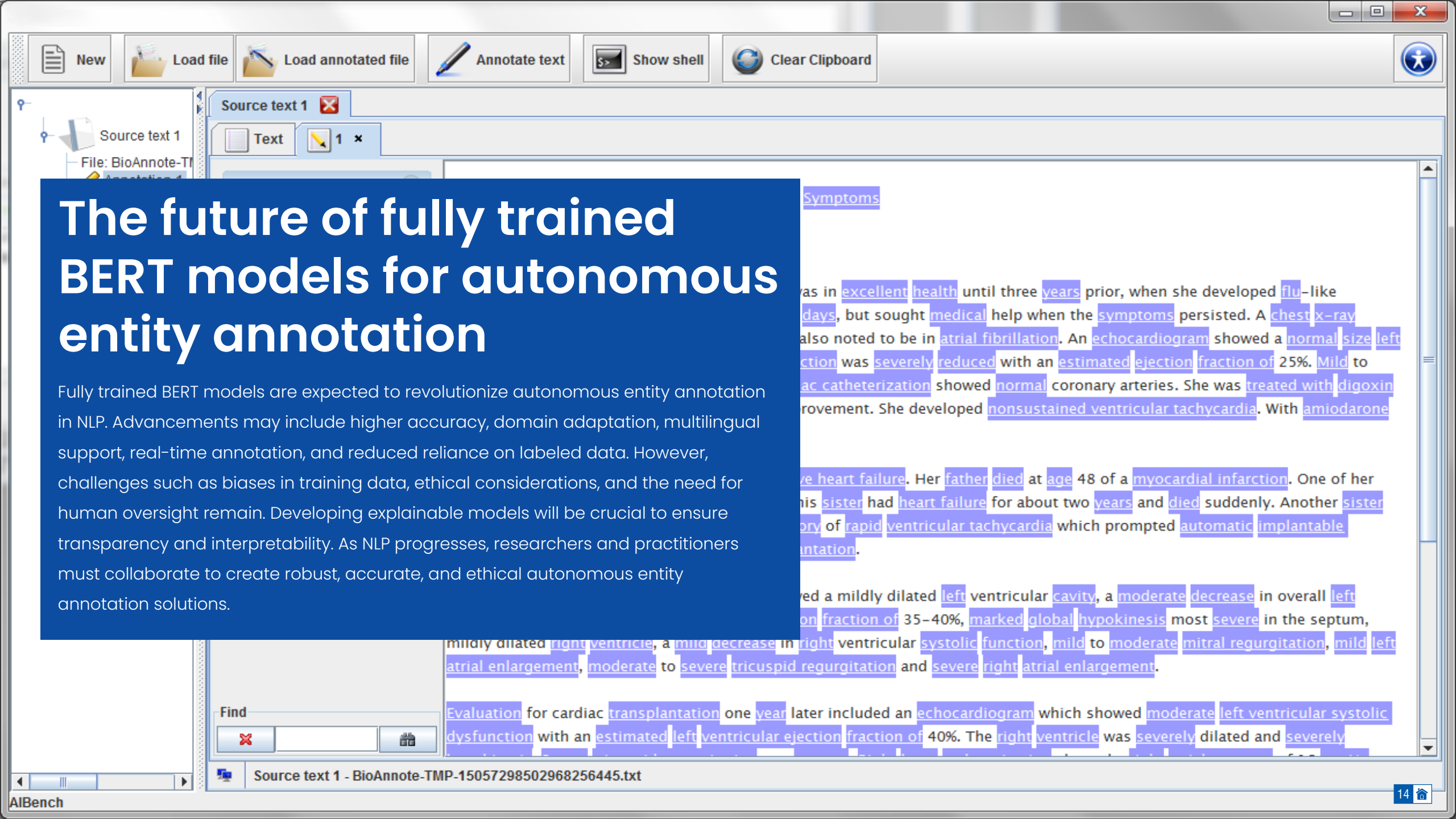
Impact and results

Our project was a success in several ways.

- The model we trained achieved high precision and recall scores, demonstrating its capability to handle complex scientific text.
- The results were integrated into the company's knowledge graph, improving the accuracy and detail of their in-house tools.
- This had a direct impact on the efficiency of their research process, facilitating more targeted, faster drug discovery initiatives.

Value addition

Our project didn't stop at delivering a one-time solution. The entity extraction model Hitech built has been deployed as a continuous learning system. As more papers are published and more data is available, the model Continues to learn and adapt, making the knowledge graph an ever-evolving resource. Through this project, we utilized the power of NLP and BERT in turning large amounts of unstructured data into actionable, structured insights.



The future of fully trained BERT models for autonomous entity annotation

Fully trained BERT models are expected to revolutionize autonomous entity annotation in NLP. Advancements may include higher accuracy, domain adaptation, multilingual support, real-time annotation, and reduced reliance on labeled data. However, challenges such as biases in training data, ethical considerations, and the need for human oversight remain. Developing explainable models will be crucial to ensure transparency and interpretability. As NLP progresses, researchers and practitioners must collaborate to create robust, accurate, and ethical autonomous entity annotation solutions.

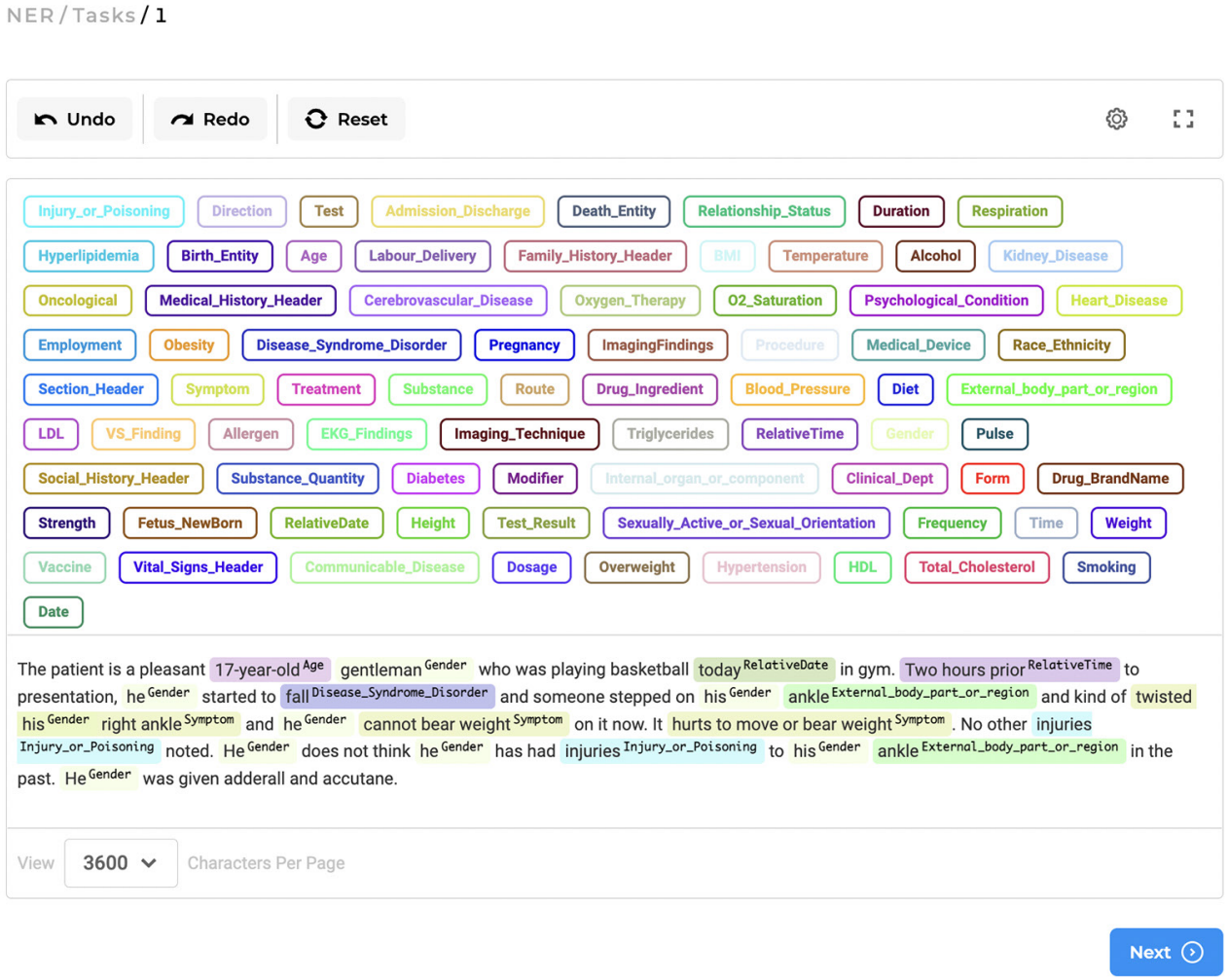
Conclusion

Leveraging BERT models in entity annotation tasks enhances the quality and performance of machine learning models. BERT's bidirectional context, pre-training, and transfer learning capabilities enable it to capture rich language representations and adapt to specific tasks with minimal labeled data.

By exploring various BERT variants tailored to different domains, languages, and computational resources, AI and ML practitioners can enhance their entity annotation projects and drive innovation across diverse industries. As the field of NLP continues to evolve, incorporating BERT models in entity annotation projects will remain a crucial strategy for achieving state-of-the-art performance and success for ML models and AI.

Unleash the true potential of your ML models. Leverage BERT for accurate and robust entity annotation.

Connect with us today »



Featured Blogs

Data Annotation for ML Projects
5 Most Effective Ways to Label Data

How Image Annotation Can Support
AI-Based Emotion Detection

hitech

Your Partner in Digital Excellence

Since 1992

www.hitechbpo.com

info@hitechbpo.com