Bringing Scientific Papers to Life
ICDAR 2023 Keynote

Marti Hearst
UC Berkeley

In collaboration with the Allen Institute of AI
Scholarly Papers Contain The World’s Scientific Knowledge

Publications are growing exponentially And papers can be challenging to understand
Scholarly Papers Contain
The World’s Scientific Knowledge

How can we make that knowledge more widely and broadly accessible?
Scholarly Papers Contain The World’s Scientific Knowledge

How can we make that knowledge more widely and broadly accessible?

Our approach: Use NLP, HCI, and document analysis to augment papers.
Talk Outline

1. Three Motivating Demos

2. AI2’s Semantic Reader Project

3. NLP and Document Analysis Techniques
   Including GPT-4 Experiments
Talk Outline

1. Three Motivating Demos

2. AI2’s Semantic Reader Project

3. NLP and Document Analysis Techniques Including GPT-4 Experiments
Talk Outline

1. Three Motivating Demos

To make scientific papers more understandable
SCHOLARPHI

Augmenting Scientific Papers with Just-in-Time, Position-Sensitive Definitions of Terms and Symbols

@ ACM CHI ’21.
What makes scientific papers challenging to read?

Linguistically-Informed Self-Attention for Semantic Role Labeling

Emma Strubell1, Patrick Verga1, Daniel Andor2, David Weiss2 and Andrew McCallum1
1College of Information and Computer Sciences
University of Massachusetts Amherst
{strubell, pat, mccallum}@cs.umass.edu
2Google AI Language
New York, NY
{andor, djweiss}@google.com

Abstract
Current state-of-the-art semantic role labeling (SRL) uses a deep neural network with no explicit linguistic features. However, prior work has shown that gold syntax trees can dramatically improve SRL decoding, suggesting the possibility of increased accuracy from explicit modeling of syntax. In this work, we present linguistically-informed self-attention (LISA): a neural network model that combines multi-head self-attention with multi-task learning across dependency parsing, part-of-speech tagging, predicate detection and SRL. Unlike previous models which require significant pre-processing to prepare linguistic features, LISA can incorporate syntax using merely raw tokens as input, encoding the sequence only once to simultaneously perform parsing, predicate detection and role labeling for all predicates. Syntax is incorporated by training one attention head to attend to syntactic parents for each token. Moreover, if a high-quality syntactic parse is already available, it can be beneficially injected at test time without re-training our SRL model.

In experiments on CoNLL-2005 SRL, LISA shown to improve results in challenging downstream tasks such as dialog systems (Tur et al., 2005; Chen et al., 2013), machine reading (Berant et al., 2014; Wang et al., 2015) and translation (Liu and Gildea, 2010; Bazrafshan and Gildea, 2013).

Though syntax was long considered an obvious prerequisite for SRL systems (Levin, 1993; Punyakanok et al., 2008), recently deep neural network architectures have surpassed syntactically-informed models (Zhou and Xu, 2015; Marcheggiani et al., 2017; He et al., 2017; Tan et al., 2018; He et al., 2018), achieving state-of-the-art SRL performance with no explicit modeling of syntax. An additional benefit of these end-to-end models is that they require just raw tokens and (usually) detected predicates as input, whereas richer linguistic features typically require extraction by an auxiliary pipeline of models.

Still, recent work (Roth and Lapata, 2016; He et al., 2017; Marcheggiani and Titov, 2017) indicates that neural network models could see even higher accuracy gains by leveraging syntactic information rather than ignoring it. He et al. (2017)
Linguistically Informed Self-Attention for Semantic Role Labeling

Emma Nekelof, Patrick Vergna, Daniel Andor, David Wieling and Andrew McCullough

University of Massachusetts Amherst

Andor, djeimse@gmail.com

Google AI Language

Abstract

Current state-of-the-art semantic role labeling (SRL) uses a deep neural network with self-attention [9] that has shown to perform well. However, the design of this network has shown that it does not naturally improve SRL, especially given the possibility of increased accuracy from reusing modeling of context. In this work, we present linguistically informed self-attention (LISA), a neural network model that incorporates information such as incorporation of linguistic features, learning across dependency parsing, past-event tagging, previous events and more. Unlike previous models which require significant pre-processing to prepare linguistic features, LISA can incorporate using ready-made features, improve the quality of predictions, and better utilize the information about context for predictions, increasing prediction quality and real-world benefit. For all predictions, features are extracted from current and past tense to aid in sentence understanding. As a result of this, LISA achieves a new state-of-the-performance for a model using pretrained features and models word embeddings, starting 2.7 F1 points higher on average.

1 Introduction

Semantic role labeling (SRL) estimates a high-level representation of meaning from a sentence, labeling which words refer to what roles in a sentence. Explicit representation of such semantic information has been shown to improve results in challenging downstream tasks such as slot filling (Tesche et al., 2005; Chou et al., 2016), ranking ranking (Boson et al., 2016; Wang et al., 2015) and translation (Lu and Cao, 2015; Barbarossa and Glinos, 2015).

Though semantic role labeling has been an active research question in NLP for over 15 years (Nenkova et al., 2006), recently deep neural network architectures have surpassed syntactically informed models (Zhao and Nen, 2015; Marcheggiani et al., 2015; He et al., 2017; Yin et al., 2018; He et al., 2018), achieving state-of-the-art SRL performance with no explicit modeling of syntax. An additional benefit of these end-to-end models is that they require just raw tokens and (usually) detected dependencies as input, whereas richer linguistic features typically require extraction by an auxiliary module of node labels.

Mean work (Hoth and Logan, 2014), He et al., 2017, Marcheggiani and Nen, 2015) indicates that neural network models could have even higher accuracy gains by leveraging semantic information rather than just pos (He et al., 2017) indicate that many of the errors made by syntax-free neural models are tied to certain semantic conditions such as propositional phrase information. Therefore, in this work, we propose a vertically low-accuracy pre-optimized model for SRL based on existing architectures. Furthermore, learning a pos-emitting policy requires to predict and adapt to the previous state of the model, and we observe that dimensions differ from the root of the attention branch to more deeply follow the com- position of Distant and Mentats (2017). Unlike the other attention mechanisms, our model learns to score dependencies in a way that can operate as a baseline for obtaining attention weights: \( \alpha_{i,j} = \frac{e^{a_{i,j}}}{\sum_{j} e^{a_{i,j}}} \) (4)

These attention weights are then multiplied by \( \phi \) for each token to obtain the self-attention injection representations \( \tilde{h}_{j}^{(n)} = \phi \cdot \sum_{i} \tilde{h}_{i} \cdot \alpha_{i,j} \) of each token \( j \). Therefore, the weighted tokens with respect to each word given \( \phi \bar{w}_{j}^{(n)} \) over the token representations in \( h_{j}^{(n)} \). The output of all attention heads for each token \( j \) are concatenated and the result is passed to the final forward layer, which consists of two (linear) projections each followed by batch REU activations (Mure et al., 2015). We will also present the use of the forward-to-the initial representation and apply layer normalization to give the final output of self-attention layer \( j \), as in Eq. 1.

2 Syntactic Informed Self-Attention

Typically, neural attention mechanisms are fed on their own to be trained to extract relevant inputs. In this study, we propose training the self-attention to attend to specific tokens corresponding to the syntactic structure of the sentence as anachronistic for parsing linguistic information to their layer.

Specifically, we replace our previous attention head with a new head that incorporates syntactic knowledge from a dependency parser (2017), trained to predict syntactic dependencies. Let \( \alpha_{i,j}^{(s)} \) be the parser-attention weights at layer \( j \). In input to the attention branch at layer \( j \), the other head receives the \( \phi \)-weighted representations of each token \( i \) and the output of the previous layer \( \phi \) (5).

As for the other attention heads, we project \( \phi \cdot h_{j}^{(n)} \) onto \( \phi \cdot h_{j}^{(n)} \cdot \alpha_{i,j} \) (5) then apply a key value and query representation decoder \( \phi \cdot h_{j}^{(n)} \). The key and query projections correspond to past and present information of the token, and we observe that dimensions differ from the root of the attention branch to more deeply follow the com- position of Distant and Mentats (2017). Unlike the other attention mechanisms, our model learns to score dependencies in a way that can operate as a baseline for obtaining attention weights: \( \alpha_{i,j} = \frac{e^{a_{i,j}}}{\sum_{j} e^{a_{i,j}}} \) (4)
Linguistically-Informed Self-Attention for Semantic Role Labelling

Emma Nsadoh*, Patrick Verge*, Daniel Andor*, David Weir† and Andrew McCarron‡

College of Information and Computer Science
University of Massachusetts Amherst

Google AI Language

abstract
Current state-of-the-art semantic role labelling (SRL) uses a deep neural network with self-attention mechanisms, and the recent paradigm shift towards language modeling has shown that pre-trained language models can dramatically improve SRL, leveraging the possibility of cross-semantic access from pre-trained models. In this work, we present linguistically-informed self-attention (LISA), a neural model that can incorporate linguistic knowledge to improve SRL performance without the learning across dependency parsing, part-of-speech tagging, previous sentences and SRL. Unlike previous models which require significant programming to propose linguistic features, LISA allows the incorporation of linguistic features using a simple and natural framework, whereas other models typically require extraction by domain-specific rules or hand-crafted features.

1 Introduction
Semantic role labelling (SRL) indicates a high-level representation of meaning between a sentence, labeling each word that takes a role. Explicit representation of such semantic information has been shown to improve results in challenging domains such as dialogue systems (Lee et al., 2005; Cohn et al., 2005), machine reading (Bosch et al., 2014), question answering (Liu and Gildea, 2010; Bui and Gildea, 2017) and reading comprehension. However, our model is the first to incorporate linguistic features using a natural framework.

[Table 1: Precision, recall and F1 on the CoNLL-2005 development and test sets]

LISA
+D&M

<table>
<thead>
<tr>
<th>Task</th>
<th>Model</th>
<th>Dev</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>P</td>
<td>F1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>W+P</td>
<td>0.85</td>
<td>0.68</td>
<td>0.71</td>
</tr>
<tr>
<td>W</td>
<td>0.84</td>
<td>0.70</td>
<td>0.73</td>
</tr>
<tr>
<td>W+M</td>
<td>0.86</td>
<td>0.72</td>
<td>0.75</td>
</tr>
</tbody>
</table>

LISA performs on the same level as existing models, achieving state-of-the-art results on the CoNLL-2005 development and test sets for both P and W+P.

4.1 Semantic role labelling

Table 1 presents precision, recall and F1 on the CoNLL-2005 development and test sets using pre-trained models for SRL. The table shows that LISA, both with and without D&M, outperforms existing models, achieving state-of-the-art results on the CoNLL-2005 development and test sets for both P and W+P.

*Equal contribution
†PhD Student
‡Assistant Professor
acronyms

symbols

LISA

+D&M

4.1 Semantic role labeling

Table 3: Precision, recall, and F1 on the CoNLL-2005 development and test sets using per-
model training. In the table, we also show the performance of our models on the same test-
sets as those used in the CoNLL-2005 shared task.

Table 4: Confusion matrix for the LISA and +D&M models on the CoNLL-2005 test set.

Table 5: The average performance of the models on the CoNLL-2005 test set, as measured by
the F1 score.

Table 6: Confusion matrix for the LISA and +D&M models on the CoNLL-2005 test set.

Table 7: The average performance of the models on the CoNLL-2005 test set, as measured by
the F1 score.

Table 8: Confusion matrix for the LISA and +D&M models on the CoNLL-2005 test set.

Table 9: The average performance of the models on the CoNLL-2005 test set, as measured by
the F1 score.

Table 10: Confusion matrix for the LISA and +D&M models on the CoNLL-2005 test set.

Table 11: The average performance of the models on the CoNLL-2005 test set, as measured by
the F1 score.

Table 12: Confusion matrix for the LISA and +D&M models on the CoNLL-2005 test set.

Table 13: The average performance of the models on the CoNLL-2005 test set, as measured by
the F1 score.

Table 14: Confusion matrix for the LISA and +D&M models on the CoNLL-2005 test set.

Table 15: The average performance of the models on the CoNLL-2005 test set, as measured by
the F1 score.

Table 16: Confusion matrix for the LISA and +D&M models on the CoNLL-2005 test set.

Table 17: The average performance of the models on the CoNLL-2005 test set, as measured by
the F1 score.

Table 18: Confusion matrix for the LISA and +D&M models on the CoNLL-2005 test set.

Table 19: The average performance of the models on the CoNLL-2005 test set, as measured by
the F1 score.

Table 20: Confusion matrix for the LISA and +D&M models on the CoNLL-2005 test set.

Table 21: The average performance of the models on the CoNLL-2005 test set, as measured by
the F1 score.

Table 22: Confusion matrix for the LISA and +D&M models on the CoNLL-2005 test set.

Table 23: The average performance of the models on the CoNLL-2005 test set, as measured by
the F1 score.
Linguistically-Informed Self-Attention for Semantic Role Labeling

Emma Nikolaidis, Patrick Verga, David Anderson, David Weir and Andrew McCallum

College of Information and Computer Science
University of Massachusetts Amherst

Google AI Language

Google AI


dev | test

Table 1: Precision, recall and F1 for the CoNLL-2005 development and test sets.

<table>
<thead>
<tr>
<th>Dev</th>
<th>P</th>
<th>R</th>
<th>F1</th>
<th>Brown Test</th>
<th>P</th>
<th>R</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>SA</td>
<td>85.9</td>
<td>85.9</td>
<td>85.9</td>
<td>86.3</td>
<td>86.3</td>
<td>86.3</td>
<td></td>
</tr>
<tr>
<td>+D&amp;M</td>
<td>88.7</td>
<td>88.7</td>
<td>88.7</td>
<td>89.3</td>
<td>89.3</td>
<td>89.3</td>
<td></td>
</tr>
</tbody>
</table>

4.1 Semantic role labeling

Table 1: Time precision, recall and F1 on the CoNLL-2005 development and test sets using pre-trained +D&M.

We train a model using 4.6% of the CoNLL-2005 development and test sets using pre-trained +D&M. This model shows an average accuracy of 89.3% on both the development and test sets, which is 2.5% higher than the best baseline model without pre-trained +D&M. This indicates that pre-trained +D&M can improve the performance of our model.

Table 3: Precision, recall and F1 on the CoNLL-2005 development and test sets using pre-trained +D&M.

We train a model using 4.6% of the CoNLL-2005 development and test sets using pre-trained +D&M. This model shows an average accuracy of 89.3% on both the development and test sets, which is 2.5% higher than the best baseline model without pre-trained +D&M. This indicates that pre-trained +D&M can improve the performance of our model.

Table 4: Precision, recall and F1 on the CoNLL-2005 development and test sets using pre-trained +D&M.

We train a model using 4.6% of the CoNLL-2005 development and test sets using pre-trained +D&M. This model shows an average accuracy of 89.3% on both the development and test sets, which is 2.5% higher than the best baseline model without pre-trained +D&M. This indicates that pre-trained +D&M can improve the performance of our model.

Table 5: Precision, recall and F1 on the CoNLL-2005 development and test sets using pre-trained +D&M.

We train a model using 4.6% of the CoNLL-2005 development and test sets using pre-trained +D&M. This model shows an average accuracy of 89.3% on both the development and test sets, which is 2.5% higher than the best baseline model without pre-trained +D&M. This indicates that pre-trained +D&M can improve the performance of our model.

Table 6: Precision, recall and F1 on the CoNLL-2005 development and test sets using pre-trained +D&M.

We train a model using 4.6% of the CoNLL-2005 development and test sets using pre-trained +D&M. This model shows an average accuracy of 89.3% on both the development and test sets, which is 2.5% higher than the best baseline model without pre-trained +D&M. This indicates that pre-trained +D&M can improve the performance of our model.

Table 7: Precision, recall and F1 on the CoNLL-2005 development and test sets using pre-trained +D&M.

We train a model using 4.6% of the CoNLL-2005 development and test sets using pre-trained +D&M. This model shows an average accuracy of 89.3% on both the development and test sets, which is 2.5% higher than the best baseline model without pre-trained +D&M. This indicates that pre-trained +D&M can improve the performance of our model.

Table 8: Precision, recall and F1 on the CoNLL-2005 development and test sets using pre-trained +D&M.

We train a model using 4.6% of the CoNLL-2005 development and test sets using pre-trained +D&M. This model shows an average accuracy of 89.3% on both the development and test sets, which is 2.5% higher than the best baseline model without pre-trained +D&M. This indicates that pre-trained +D&M can improve the performance of our model.

Table 9: Precision, recall and F1 on the CoNLL-2005 development and test sets using pre-trained +D&M.

We train a model using 4.6% of the CoNLL-2005 development and test sets using pre-trained +D&M. This model shows an average accuracy of 89.3% on both the development and test sets, which is 2.5% higher than the best baseline model without pre-trained +D&M. This indicates that pre-trained +D&M can improve the performance of our model.

Table 10: Precision, recall and F1 on the CoNLL-2005 development and test sets using pre-trained +D&M.

We train a model using 4.6% of the CoNLL-2005 development and test sets using pre-trained +D&M. This model shows an average accuracy of 89.3% on both the development and test sets, which is 2.5% higher than the best baseline model without pre-trained +D&M. This indicates that pre-trained +D&M can improve the performance of our model.

Table 11: Precision, recall and F1 on the CoNLL-2005 development and test sets using pre-trained +D&M.

We train a model using 4.6% of the CoNLL-2005 development and test sets using pre-trained +D&M. This model shows an average accuracy of 89.3% on both the development and test sets, which is 2.5% higher than the best baseline model without pre-trained +D&M. This indicates that pre-trained +D&M can improve the performance of our model.

Table 12: Precision, recall and F1 on the CoNLL-2005 development and test sets using pre-trained +D&M.

We train a model using 4.6% of the CoNLL-2005 development and test sets using pre-trained +D&M. This model shows an average accuracy of 89.3% on both the development and test sets, which is 2.5% higher than the best baseline model without pre-trained +D&M. This indicates that pre-trained +D&M can improve the performance of our model.

Table 13: Precision, recall and F1 on the CoNLL-2005 development and test sets using pre-trained +D&M.

We train a model using 4.6% of the CoNLL-2005 development and test sets using pre-trained +D&M. This model shows an average accuracy of 89.3% on both the development and test sets, which is 2.5% higher than the best baseline model without pre-trained +D&M. This indicates that pre-trained +D&M can improve the performance of our model.

Table 14: Precision, recall and F1 on the CoNLL-2005 development and test sets using pre-trained +D&M.

We train a model using 4.6% of the CoNLL-2005 development and test sets using pre-trained +D&M. This model shows an average accuracy of 89.3% on both the development and test sets, which is 2.5% higher than the best baseline model without pre-trained +D&M. This indicates that pre-trained +D&M can improve the performance of our model.

Table 15: Precision, recall and F1 on the CoNLL-2005 development and test sets using pre-trained +D&M.

We train a model using 4.6% of the CoNLL-2005 development and test sets using pre-trained +D&M. This model shows an average accuracy of 89.3% on both the development and test sets, which is 2.5% higher than the best baseline model without pre-trained +D&M. This indicates that pre-trained +D&M can improve the performance of our model.

Table 16: Precision, recall and F1 on the CoNLL-2005 development and test sets using pre-trained +D&M.

We train a model using 4.6% of the CoNLL-2005 development and test sets using pre-trained +D&M. This model shows an average accuracy of 89.3% on both the development and test sets, which is 2.5% higher than the best baseline model without pre-trained +D&M. This indicates that pre-trained +D&M can improve the performance of our model.

Table 17: Precision, recall and F1 on the CoNLL-2005 development and test sets using pre-trained +D&M.

We train a model using 4.6% of the CoNLL-2005 development and test sets using pre-trained +D&M. This model shows an average accuracy of 89.3% on both the development and test sets, which is 2.5% higher than the best baseline model without pre-trained +D&M. This indicates that pre-trained +D&M can improve the performance of our model.

Table 18: Precision, recall and F1 on the CoNLL-2005 development and test sets using pre-trained +D&M.

We train a model using 4.6% of the CoNLL-2005 development and test sets using pre-trained +D&M. This model shows an average accuracy of 89.3% on both the development and test sets, which is 2.5% higher than the best baseline model without pre-trained +D&M. This indicates that pre-trained +D&M can improve the performance of our model.

Table 19: Precision, recall and F1 on the CoNLL-2005 development and test sets using pre-trained +D&M.

We train a model using 4.6% of the CoNLL-2005 development and test sets using pre-trained +D&M. This model shows an average accuracy of 89.3% on both the development and test sets, which is 2.5% higher than the best baseline model without pre-trained +D&M. This indicates that pre-trained +D&M can improve the performance of our model.

Table 20: Precision, recall and F1 on the CoNLL-2005 development and test sets using pre-trained +D&M.

We train a model using 4.6% of the CoNLL-2005 development and test sets using pre-trained +D&M. This model shows an average accuracy of 89.3% on both the development and test sets, which is 2.5% higher than the best baseline model without pre-trained +D&M. This indicates that pre-trained +D&M can improve the performance of our model.

Table 21: Precision, recall and F1 on the CoNLL-2005 development and test sets using pre-trained +D&M.

We train a model using 4.6% of the CoNLL-2005 development and test sets using pre-trained +D&M. This model shows an average accuracy of 89.3% on both the development and test sets, which is 2.5% higher than the best baseline model without pre-trained +D&M. This indicates that pre-trained +D&M can improve the performance of our model.

Table 22: Precision, recall and F1 on the CoNLL-2005 development and test sets using pre-trained +D&M.

We train a model using 4.6% of the CoNLL-2005 development and test sets using pre-trained +D&M. This model shows an average accuracy of 89.3% on both the development and test sets, which is 2.5% higher than the best baseline model without pre-trained +D&M. This indicates that pre-trained +D&M can improve the performance of our model.

Table 23: Precision, recall and F1 on the CoNLL-2005 development and test sets using pre-trained +D&M.

We train a model using 4.6% of the CoNLL-2005 development and test sets using pre-trained +D&M. This model shows an average accuracy of 89.3% on both the development and test sets, which is 2.5% higher than the best baseline model without pre-trained +D&M. This indicates that pre-trained +D&M can improve the performance of our model.
Acronyms

- AI
- ANN
- CNN
- DNN
- LDA
- SVM
- ML
- NLP
- NN

Symbols

- \(\phi\)
- \(\lambda\)
- \(\theta\)
- \(\alpha\)
- \(\beta\)

New Terms

- Information Retrieval (IR)
- Natural Language Processing (NLP)
- Machine Learning (ML)
- Deep Learning (DL)
- Reinforcement Learning (RL)

ScholarPhi

an augmented reading interface that surfaces just-in-time, position-sensitive definitions of nonce words.

See demo video at https://youtu.be/yYcQf-Yq8B0
ScholarPhi

an augmented reading interface that surfaces just-in-time, position-sensitive definitions of nonce words.
Simplifying the Complex

Paper Plain: Making Medical Research Papers Approachable to Healthcare Consumers with Natural Language Processing

@ ACM TOCHI ’23.
Therapeutic peptides for the treatment of systemic lupus erythematosus: a place in therapy

Rossella Talotta*, Fabiola Atzeni* and Magdalena Janina Laska

*Department of Clinical and Experimental Medicine, Rheumatology Unit, Azienda Ospedaliera "Gaetano Martino", University of Messina, Messina, Italy; *Department of Rheumatology, Aarhus University Hospital, Aarhus, Denmark

ABSTRACT

Introduction: Studies in vitro and in vivo have identified several peptides that are potentially useful in treating systemic lupus erythematosus (SLE). The rationale for their use lies in the cost-effective production, high potency, target selectivity, low toxicity, and a peculiar mechanism of action that is mainly based on the induction of immune tolerance. Three therapeutic peptides have entered clinical development, but they have yielded disappointing results. However, some subsets of patients, such as those with the positivity of anti-dsDNA antibodies, appear more likely to respond to these medications.

Areas covered: This review evaluates the potential use of therapeutic peptides for SLE and gives an opinion on how they may offer advantages for SLE treatment.

Expert opinion: Given their acceptable safety profile, therapeutic peptides could be added to agents traditionally used to treat SLE and this may offer a synergistic and drug-sparing effect, especially in selected patient populations. Moreover, they could temporally be utilized to manage SLE flares, or be administered as a vaccine in subjects at risk. Efforts to ameliorate bioavailability, increase the half-life and prevent immunogenicity are ongoing. The formulation of hybrid compounds, like peptibodies or peptidomimetic small molecules, is expected to yield renewed treatments with a better pharmacologic profile and increased efficacy.

Introduction

Systemic lupus erythematosus (SLE) is the prototypical autoimmune connective tissue disease, affecting 5 million individuals worldwide, mainly women during the fertile age [1]. Clinical presentation broadly varies from patient to patient, with kidney and central nervous system (CNS) involvement representing the most severe complications [2]. The disease and usually shows less successful results, Figure 2 [8]. Several immunologic pathways are, in fact, concomitantly activated in SLE, and this justifies the use of medications like steroids, immunosuppressants and disease-modifying anti-rheumatic drugs (DMARDs), which unselectively counteract the immune response. Such a combo-therapy can indeed have many summing side effects that can be further
Therapeutic peptides for the treatment of systemic lupus erythematosus: a place in therapy

Rossella Talotta, Fabiola Atzeni, and Magdalena Janina Laska

*Department of Clinical and Experimental Medicine, Rheumatology Unit, Azienda Ospedaliera “Gaetano Martino”, University of Messina, Messina, Italy; ‡Department of Rheumatology, Aarhus University Hospital, Aarhus, Denmark

ABSTRACT
Introduction: Studies in vitro and in vivo have identified several peptides that are potentially useful in treating systemic lupus erythematosus (SLE). The rationale for their use lies in the cost-effective production, high potency, target selectivity, low toxicity, and a peculiar mechanism of action that is mainly based on the induction of immune tolerance. Three therapeutic peptides have entered clinical development, but they have yielded disappointing results. However, some subsets of patients, such as those with the positivity of anti-dsDNA antibodies, appear more likely to respond to these medications.

Areas covered: This review evaluates the potential use of therapeutic peptides for SLE and gives an opinion on how they may offer advantages for SLE treatment.

Expert opinion: Given their acceptable safety profile, therapeutic peptides could be added to agents traditionally used to treat SLE and this may offer a synergistic and drug-sparing effect, especially in selected patient populations. Moreover, they could temporally be utilized to manage SLE flares, or be administered as a vaccine in subjects at risk. Efforts to ameliorate bioavailability, increase the half-life, and prevent immunogenicity are ongoing. The formulation of hybrid compounds, like peptibodies or peptidomimetic small molecules, is expected to yield renewed treatments with a better pharmacologic profile and increased efficacy.

. Introduction

Systemic lupus erythematosus (SLE) is the prototypical autoimmune connective tissue disease, affecting 5 million individuals worldwide, mainly women during the fertile age [1]. Clinical presentation broadly varies from patient to patient, with kidney and central nervous system (CNS) involvement representing the most severe complications [2]. The disease usually shows less successful results. Figure 2 [8]. Several immunologic pathways are, in fact, concomitantly activated in SLE, and this justifies the use of medications like steroids, immunosuppressants and disease-modifying anti-rheumatic drugs (DMARDs), which unelectively counteract the immune response. Such a combo-therapy can indeed have many summing side effects that can be further
To help you read this paper, below is a list of key questions that are answered in this paper.

What condition does this paper study?
Systemic Lupus Erythematosus is a disease that affects about 5 million people in the world, mostly women. see paragraph on page 2

How is the condition usually treated?
After you get the diagnosis of lupus, the doctor will see how bad your lupus is and how much it affects your body. see paragraph on page 5

EXPERT OPINION ON INVESTIGATIONAL DRUGS
2020, VOL. 29, NO. 8, 845–867
https://doi.org/10.1080/13543784.2020.1777983

REVIEW

Therapeutic peptides for the treatment of systemic lupus erythematosus therapy

Rossella Talotta, Fabiola Atzeni and Magdalena Janina Laska

University Hospital, Aarhus, Denmark

aDepartment of Clinical and Experimental Medicine, Rheumatology Unit, Azienda Ospedaliera Universitaria, Italy; bDepartment of Rheumatology, Aarhus University Hospital, Aarhus, Denmark

ABSTRACT

Introduction: Studies in vitro and in vivo have identified several peptides that can treat systemic lupus erythematosus (SLE). The rationale for their use is their high production, high potency, target selectivity, low toxicity, and a peculiar mechanism of action mainly based on the induction of immune tolerance. Three therapeutic peptide candidates have reached the development stage, but they have yielded disappointing results. However, some studies with the positivity of anti-dsDNA antibodies, appear more likely to respond to these therapies. The latest studies have focused on the relationship between immune response and the development of lupus nephritis, as well as the identification of new targets for the treatment of lupus.

Areas covered: This review evaluates the potential use of therapeutic peptides and their opinion on how they may offer advantages for SLE treatment.

Expert opinion: Given their acceptable safety profile, therapeutic peptides can be used as an alternative to traditionally used treatments for SLE and may offer a synergistic and drug-specific approach to the treatment of SLE.
Therapeutic peptides for the treatment of systemic lupus erythematosus: a place in therapy

Rossella Talotta, Fabiola Atzeni and Magdalena Janina Laska

*Department of Clinical and Experimental Medicine, Rheumatology Unit, Azienda Ospedaliera “Gaetano Martino”, University of Messina, Messina, Italy; **Department of Rheumatology, Aarhus University Hospital, Aarhus, Denmark

ABSTRACT

Introduction: Studies in vitro and in vivo have identified several peptides that are potentially useful in treating systemic lupus erythematosus (SLE). The rationale for their use lies in the cost-effective production, high potency, target selectivity, low toxicity, and a peculiar mechanism of action that is mainly based on the induction of immune tolerance. Three therapeutic peptides have entered clinical development, but they have yielded disappointing results. However, some subsets of patients, such as those with the positivity of anti-dsDNA antibodies, appear more likely to respond to these medications.

Areas covered: This review evaluates the potential use of therapeutic peptides for SLE and gives an opinion on how they may offer advantages for SLE treatment.

Expert opinion: Given their acceptable safety profile, therapeutic peptides could be added to agents traditionally used to treat SLE and this may offer a synergistic and drug-sparing effect, especially in selected patient populations. Moreover, they could temporarily be utilized to manage SLE flares, or be administered as a vaccine in subjects at risk. Efforts to ameliorate bioavailability, increase the half-life and prevent immunogenicity are ongoing. The formulation of hybrid compounds, like peptibodies or peptidomimetic small molecules, is expected to yield renewed treatments with a better pharmacologic profile and increased efficacy.

Introduction

Systemic lupus erythematosus (SLE) is the prototypical autoimmune connective tissue disease, affecting 5 million individuals worldwide, mainly women during the fertile age [1]. Clinical presentation broadly varies from patient to patient, with kidney and central nervous system (CNS) involvement representing the most severe complications [2]. The disease and usually shows less successful results, Figure 2 [8]. Several immunologic pathways are, in fact, concomitantly activated in SLE, and this justifies the use of medications like steroids, immunosuppressants and disease-modifying anti-rheumatic drugs (DMARDs), which unselectively counteract the immune response. Such a combo-therapy can indeed have many summing side effects that can be further...
**Conclusion**

Better knowledge of the pathogenesis of SLE is expected to enrich the therapeutic armamentarium and facilitate the management of the disease. The use of peptides, specifically designed to target SLE-related epitopes or crucial pathways, may represent a novel fascinating opportunity. Given their good safety profile and immunomodulatory properties, therapeutic peptides could be added to standard of care, and, perhaps, allow the sparing of conventional drugs. In addition, their prescription might be tailored to specific subsets of patients having the highest likelihood of response. Nevertheless, despite the successful results observed in preclinical studies, RCTs showed a controversial efficacy profile concerning the use of these compounds in SLE. It is expected that future research, aiming at the amelioration of their physicochemical properties and at the improvement in the design of clinical trials, will bring more encouraging data on this innovative therapeutic panorama.

**Expert opinion**

The treatment of SLE still relies on a conservative approach, combining multiple unselective immunosuppressive agents [8] and, consecutively, increasing the risk of unwanted side effects. Unlike other rheumatic diseases, the licensed use of biologic agents, which electively inhibit a specific target, has been solely limited to belimumab. Rituximab failed to achieve the primary endpoints in RCTs conducted in SLE patients [12,13,143] but, due to encouraging real-life data [11], its off-label use is advised in resistant severe manifestations [8]. The potential use of novel biological agents and small molecules may be translated in humans is an uncertain issue that needs to be addressed in future research.

The discovery of novel molecular targets is expected to enrich, in the next years, the panorama of therapeutic peptides for SLE, among which CXCR4 and STING antagonists and virus-derived peptides seem promising candidates [116,119,133].

Meanwhile, research is focusing on the optimization of the physicochemical structure of preexisting peptides with the aim of improving their pharmacologic properties, including bioavailability and half-life. One of these efforts consists of the
release of IL-6 and IFNγ following the stimulation with mitogens in PBMCs of patients with SLE. Additionally, when subcutaneously given at a dose of 10–100 mg/kg 5 days a week to MRL/lpr mice, it increased life-span and reduced proteinuria, though the titers of anti-DNA antibodies and the histological grading of glomerulonephritis remained unchanged. Although promising, research in this field was discontinued.

2.3. Peptibodies

Section summary: A medicine called AMG623 is a drug that is made to treat a disease called lupus. The medicine is made of four pieces of a protein that binds to BlYs, which is a protein that makes lupus worse. The medicine is made of pieces of a protein that binds to BlYs, and it's attached to the Fc part of an antibody.

3.1. Blysibimod

AMG623 is a peptibody carrying four high-affinity BlYs-binding peptides grafted onto the Fc portion of a IgG, [97]. The compound was electively designed for SLE on the basis of the results of a phage display library screening, yielding a group of peptides binding to both soluble and membrane-bound BlYs with high affinity [98]. BlYs plays a crucial role in SLE pathogenesis, as it is able to foment B cell auto-reactivity, being also strictly interconnected with the type I IFN pathway. In preclinical experiments [99], the compound constrained the number of peripheral and spleen B lymphocytes in healthy BALB/c mice, and these results were confirmed in a BlYs-mediated B lymphocyte proliferation assay. Using NZBxNZW
Usability study

Participants: 24 non-expert readers.

Reading task: Read a paper for 10 minutes. Then, answer a set of questions about the paper.

Conditions (×2): Each reader completed tasks with 2 of 4 interface variants (one variant was a featureless baseline).
Usability Study Results

Baseline PDF reader: Most participants read papers linearly and spent substantial time in dense sections with limited important information.

Paper Plain: All participants reached the end of the paper.

Section gists and term definitions helped read dense passages.

The questions & answer gists supported quick navigation and helped decide which sections to read.
The effect on understanding papers

Participants reported it was **less difficult** to read the paper and that they **better understood** the paper, when using Paper Plain.

That said, participants answered questions about the paper **about as correctly** with both Paper Plain and the baseline. They tended to answer more correctly when questions were answerable by clicking questions in the key question index.
INTELLIGENT PAPER SCANNING

SCIM: Intelligent Skimming Support

@ ACM IUI ‘23
Lots of content – where to start reading?
Idea: Highlight Key Passages by Rhetorical Purpose

Chong Chen†, Fei Sun†, Min Zhang†, Bolin Ding‡

†Department of Computer Science and Technology, Institute for Artificial Intelligence,
Beijing National Research Center for Information Science and Technology, Tsinghua University, Beijing 100084, China
‡DAMO Academy, Alibaba Group

cchen7@mails.tsinghua.edu.cn, z-mt@tsinghua.edu.cn, 1jofey.sf@bolin.ding.ali.com

ABSTRACT

Recommender systems provide essential web services by learning users’ personal preferences from collected data. However, in many cases, systems also need to forget some training data. From the perspective of privacy, users desire a tool to erase the impacts of their sensitive data from the trained models. From the perspective of utility, if a system’s utility is damaged by some bad data, the system needs to forget such data to regain utility. While unlearning is very important, it has not been well-considered in existing recommender systems. Although there are some researches have studied the problem of machine unlearning, existing methods cannot be directly applied to recommendation as they are unable to consider the collaborative information.

In this paper, we propose RecEraser, a general and efficient machine unlearning framework tailored to recommendation tasks. The main idea of RecEraser is to divide the training set into multiple shards and train sub-models with these shards. Specifically, to keep the collaborative information of the data, we first design three novel data partition algorithms to divide training data into balanced groups. We then further propose an adaptive aggregation method to improve the global model utility. Experimental results on three public benchmarks show that RecEraser can not only achieve efficient unlearning but also outperform the state-of-the-art unlearning methods in terms of model utility. The source code can be found at https://github.com/chencnghong/Thesis/Recommendation-Unlearning

CCS CONCEPTS

• Information systems → Recommender systems; • Security and privacy → Privacy protections.

KEYWORDS

Machine Unlearning; Selective Deletion; Recommender Systems; Collaborative Filtering.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Permissions@acm.org.
Recommendation Unlearning

Chong Chen\textsuperscript{1*}, Fei Sun\textsuperscript{2T}, Min Zhang\textsuperscript{1T}, Bolin Ding\textsuperscript{2}
\textsuperscript{1}Department of Computer Science and Technology, Institute for Artificial Intelligence, Beijing National Research Center for Information Science and Technology, Tsinghua University, Beijing 100084, China
\textsuperscript{2}DAMO Academy, Alibaba Group
cc17@mails.tsinghua.edu.cn, z-m@tsinghua.edu.cn, \textsuperscript{2}feiy.sf, bolin.ding@alibaba-inc.com

ABSTRACT

Recommender systems provide essential web services by learning users’ personal preferences from collected data. However, in many cases, systems also need to forget some training data. From the perspective of privacy, users desire a tool to erase the impacts of their sensitive data from the trained models. From the perspective of utility, if a system’s utility is damaged by some bad data, the system needs to forget such data to regain utility. While unlearning is very important, it has not been well-considered in existing recommender systems. Although there are some researches have studied the problem of machine unlearning, existing methods can not be directly applied to recommendation as they are unable to consider the collaborative information.

In this paper, we propose RecEraser, a general and efficient machine unlearning framework tailored to recommendation tasks. The main idea of RecEraser is to divide the training set into multiple shards and train submodels with these shards. Specifically, to keep the collaborative information of the data, we first design three novel data partition algorithms to divide training data into balanced groups. We then further propose an adaptive aggregation method to improve the global model utility. Experimental results on three public benchmarks show that RecEraser can not only achieve efficient unlearning but also outperform the state-of-the-art unlearning methods in terms of model utility. The source code can be found at https://github.com/chenchongthu/Recommendation-Unlearning

ACM Reference Format:

1 INTRODUCTION

Recommender systems provide personalized service for users to alleviate the information overload problem, playing a more and more important role in a wide range of applications, such as e-commerce \[\text{[34, 38]}, \] social media \[\text{[8, 10, 36, 48]}, \] and news portal \[\text{[41]}, \]. The key of personalized recommender systems is known as collaborative filtering \[\text{[28, 31, 37, 42]}, \] which learns users’ preference based on their historical records (e.g., views, clicks, and ratings).

Once a recommender system is built, it has potentially memorized the training data. However, in many cases, a recommender system also needs to forget certain sensitive data and its complete lineage, which is called Recommendation Unlearning in this paper. Consider privacy first, recent researches have shown that users’ sensitive information could be leaked from the trained models, e.g., recommender systems \[\text{[50]}, \] big pretrained \[\text{[4]}, \] and finetuned natural language models \[\text{[49]}, \]. In such cases, users desire a tool to erase the impacts of their sensitive information from the trained models. The second reason is utility. Nowadays, new data is collected incrementally to further refine existing models \[\text{[51]}, \]. However, bad data (or called dirty data), e.g., polluted data in poisoning attacks \[\text{[33]}, \]
**Recommendation Unlearning**

Chen Zing*, Fei Sun‡, Min Zhang†, Bolin Ding‡

Computer Science and Technology, Institute for Artificial Intelligence,
Tsinghua University, Beijing 100084, China

‡DAMO Academy, Alibaba Group

z-m@tsinghua.edu.cn, ofey.sf, bolin.ding@alibaba-inc.com

1 INTRODUCTION

Recommender systems provide personalized service for users to alleviate the information overload problem, playing a more and more important role in a wide range of applications, such as e-commerce [34, 38], social media [8, 10, 16, 48], and news portals [41]. The key of personalized recommender systems is known as collaborative filtering [28, 31, 37, 42], which learns users' preference based on their historical records (e.g., views, clicks, and ratings).

Once a recommender system is built, it has potentially memorized the training data. However, in many cases, a recommender system also needs to forget certain sensitive data and its complete lineage, which is called Recommendation Unlearning in this paper. Consider privacy first, recent researches have shown that users' sensitive information could be leaked from the trained models, e.g., recommender systems [50], big pretrained [4] and finetuned natural language models [49]. In such cases, users desire a tool to erase the impacts of their sensitive information from the trained models. The second reason is utility. Nowadays, new data is collected incrementally to further refine existing models [51]. However, bad data (or called dirty data), e.g., polluted data in poisoning attacks [33]
Recommendation Unlearning

Chong Chen\textsuperscript{1*}, Fei Sun\textsuperscript{1†}, Min Zhang\textsuperscript{1††}, Bolin Ding\textsuperscript{2}  
\textsuperscript{1}Department of Computer Science and Technology, Institute for Artificial Intelligence, Beijing National Research Center for Information Science and Technology, Tsinghua University, Beijing 100084, China  
\textsuperscript{2}DAMO Academy, Alibaba Group  
cc17@mails.tsinghua.edu.cn, z-m@tsinghua.edu.cn, \textsuperscript{†}feiyi.f, bolin.dingl@alibaba-inc.com

\textbf{ABSTRACT}  
Recommender systems provide essential web services by learning users’ personal preferences from collected data. However, in many cases, systems also need to forget some training data. From the perspective of privacy, users desire a tool to erase the impacts of their sensitive data from the trained models. From the perspective of utility, if a system’s utility is damaged by some bad data, the system needs to forget such data to regain utility. While unlearning is very important, it has not been well-considered in existing recommender systems. Although there are some researches have studied the problem of machine unlearning, existing methods cannot be directly applied to recommendation as they are unable to consider the collaborative information.

In this paper, we propose RecEraser, a general and efficient machine unlearning framework tailored to recommendation tasks. The main idea of RecEraser is to divide the training set into multiple shards and train submodels with these shards. Specifically, to keep the collaborative information of the data, we first design three novel data partition algorithms to divide training data into balanced groups. We then further propose an adaptive aggregation method to improve the global model utility. Experimental results on three public benchmarks show that RecEraser can not only achieve efficient unlearning but also outperform the state-of-the-art unlearning methods in terms of model utility. The source code can be found at https://github.com/chengonghu/Recommendation-Unlearning

\textbf{ACM Reference Format:}  

\section{INTRODUCTION}  
Recommender systems provide personalized service for users to alleviate the information overload problem, playing a more and more important role in a wide range of applications, such as e-commerce \cite{24, 38}, social media \cite{4, 10, 16, 48}, and news portal \cite{41}. The key of personalized recommender systems is known as collaborative filtering \cite{28, 31, 37, 42}, which learns users’ preference based on their historical records (e.g., views, clicks, and ratings).

Once a recommender system is built, it has potentially memorized the training data. However, in many cases, a recommender system also needs to forget certain sensitive data and its complete lineage, which is called Recommendation Unlearning in this paper. Consider privacy-first, recent researches have shown that users’ sensitive information could be leaked from the trained models, e.g., recommender systems \cite{50}, big pretrained \cite{43} and finetuned natural language models \cite{49}. In such cases, users desire a tool to erase the impacts of their sensitive information from the trained models. The second reason is utility. Nowadays, new data is collected incrementally to further refine existing models \cite{51}. However, bad data (or called dirty data), e.g., polluted data in poisoning attacks \cite{33} or out-of-distribution (OOD) data \cite{3}, will seriously degrade the performance of recommendation. Once these data are identified, the system needs to forget them to regain utility. Moreover, generally users’ preferences are dynamic and changeable \cite{44}. For example, a user who wants to buy a mobile phone will be happy to see recommendations about mobile phones. But after the purchase, she/he will not be interested in the recommendations of new mobile phones for a period of time. In this case, the user will want to forget the previous purchase history to avoid receiving irrelevant recommendations.

\section{RELATED WORK}  

\subsection{General Unlearning}  

A few existing works have been proposed to handle the unlearning problem. The general approach for unlearning is to remove or hide data entries from the local database. However, this approach does not consider the global model utility. To the best of our knowledge, the only work that addresses the unlearning problem in the recommendation domain is RecForget \cite{1, 27}. RecForget is a model-agnostic unlearning method that removes part of the model. However, it only considers the model-agnostic utility and does not consider the collaborative information.

\subsection{Model-agnostic Unlearning}  

Recent work \cite{28} also designed a method to unlearn personal data from the model. However, it requires user consent. Furthermore, their method does not consider the global model utility.

\subsection{Adaptive Unlearning}  

In this paper, we propose RecEraser, a general and efficient unlearning framework for recommendation tasks. The main idea is to divide the training set into multiple shards and train submodels with these shards. Specifically, to keep the collaborative information of the data, we first design three novel data partition algorithms to divide training data into balanced groups. We then further propose an adaptive aggregation method to improve the global model utility.
**Usability Studies**

**In-lab study:** 19 participants, expertise in NLP
Within-participants; each used a plain PDF reader and Scim
1 hour duration

**Longitudinal Diary Study:** 12 participants, with expertise
They choose which version of the reader to use
2 weeks
In-lab Usability Study Results

Questions answered more quickly with Scim; No difference in accuracy
Most did not find highlights distracting
More akin to “scanning” than “skimming”
Longitudinal Usability Study Results

70% reported finding highlights useful
Assisted in focusing on important information
Provided a summary of the paper
Distracting when highlights are incorrect
Three Intelligently Interactive Interfaces

Because all use automation, they are going to make errors!
Doing this all on PDFs ...

Why? HTML would be easier, but PDFs are where the readers are (for now)

How? The rest of this talk.
Talk Outline

1. Three Motivating Demos

2. AI2’s Semantic Reader Project

3. NLP and Document Analysis Techniques
   Including GPT-4 Experiments
Allen Institute for AI (AI2)

A non-profit research institute dedicated to AI for the Common Good
About Semantic Scholar

Helping Scholars Discover New Insights

Semantic Scholar provides free, AI-driven search and discovery tools, and open resources for the global research community.

We index over 200 million academic papers sourced from publisher partnerships, data providers, and web crawls.
50 person team
7 year project
207M+ scientific paper index
8M+ monthly active users
PubLayNet: Largest Dataset Ever for Document Layout Analysis
Xu Zhong, Jianbin Tang, Antonio Jimeno-Yepes · Computer Science · IEEE International Conference on Document Analysis and Recognition · 16 August 2019
TLDR The PubLayNet dataset for document layout analysis is developed by automatically matching the XML representations and the content of over 1 million PDF articles that are publicly available on PubMed Central and demonstrated that deep neural networks trained on Pub LayNet accurately recognize the layout of scientific articles. Expand

Doc-GCN: Heterogeneous Graph Convolutional Networks for Document Layout Analysis
Siwen Luo, Yi Ding, Siyu Long, S. Han, Josiah Poole · Computer Science · International Conference on Computational Linguistics · 22 August 2022
TLDR The Doc-GCN presents an effective way to harmonize and integrate heterogeneous aspects for Document Layout Analysis, and achieves state-of-the-art results on three widely used DLA datasets: PubLayNet, FUNSD, and DocBank. Expand
Semantic Scholar API

Providing a reliable source of scholarly data for developers

Build projects that accelerate scientific progress with the Semantic Scholar Academic Graph API

- Code Examples
- Q&A Bank
- Request an API Key
- API Service Status Page
- Join
- Read it

Access Our Continually Updating Corpus

- 206 Million Papers
- 2.49 Billion Citations
- 79 Million Authors

https://www.semanticscholar.org/product/api
Semantic Reader Project

Introducing Semantic Reader

An AI-Powered Augmented Scientific Reading Application

https://www.semanticscholar.org/product/semantic-reader
Published as a conference paper at ICLR 2020

Semantic Reader In Action

These solutions address the memory limitation problem, but not the communication overhead. In this paper, we address all of the aforementioned problems, by designing A Lite BERT (ALBERT) architecture that has significantly fewer parameters than a traditional BERT architecture.

ALBERT incorporates two parameter reduction techniques that lift the major obstacles in scaling pre-trained models. The first one is a factorized embedding parameterization. By decomposing the large vocabulary embedding matrix into two smaller matrices, we separate the size of the hidden layers from the size of vocabulary embedding. This separation makes it easier to grow the hidden size without significantly increasing the parameter size of the vocabulary embeddings. The second technique is cross-layer parameter sharing. This technique prevents the parameter from growing with the depth of the network. Both techniques significantly reduce the number of parameters for BERT without seriously hurting performance, thus improving parameter-eficiency. An ALBERT configuration similar to BERT-large has 18x fewer parameters and can be trained about 1.7x faster. The parameter reduction techniques also act as a form of regularization that stabilizes the training and helps with generalization.

To further improve the performance of ALBERT, we also introduce a self-supervised loss for sentence-order prediction (SOP). SOP primary focuses on inter-sentence coherence and is designed to address the ineffectiveness (Yang et al., 2019; Liu et al., 2019) of the next sentence prediction (NSP) loss proposed in the original BERT. As a result of these design decisions, ALBERT that still have fewer parameters than other models can establish new state-of-the-art results for natural language understanding. SQuAD benchmark to 89.4, and the F1 score of the GLUE benchmark to 68.8.

2 RELATED WORK
2.1 SCALING UP REPRESENTATIONS

Learning representations of natural language has been widely adopted in NLP.

Currently only works on latex-backed PDFs
But ... it is a challenge to translate some of this research into the product

https://www.semanticscholar.org/product/semantic-reader
Talk Outline

1. Three Motivating Examples of UIs

2. AI2’s Semantic Reader Project

3. NLP and Document Analysis Techniques Including GPT-4 Experiments
The construction of ScholarPhi

Major challenge: fine-grained locations of symbols
The construction of ScholarPhi

Major challenge: fine-grained locations of symbols

Solution: Link latex code to location in a PDF image
Identifying bounding boxes for symbols

\text{LaTeX} \quad \text{Images of PDF}

\text{function:} P(y_t^{prp}, \text{ where...}

\text{function:} \{\textcolor{orange}{P(y_t^{prp}, \text{ where...}}

\text{Function: } P(y_l^{\text{prp}} | X) \propto \exp(\tau_l), \text{ where...}

\text{Function: } P(y_l^{\text{prp}} | X) \propto \exp(\tau_l)

P(y_l^{\text{prp}} | X) \propto \exp(\tau_l)
Identifying bounding boxes for symbols

Equation

\[ E_{f_d} = x^n \]

Contains 8 symbols

\[ E_{f_d} = x^n \]

Precision: 96%
Recall: 88%
The construction of ScholarPhi

Major challenge: fine-grained locations of symbols

Solution: Link latex code to location in a PDF image

Problems:
1) Difficult to get high accuracy efficiently
2) Does not work on PDFs without latex
The construction of ScholarPhi

Major challenge: definition recognition
The construction of ScholarPhi

Major challenge: definition recognition

Solution: new algorithm for definition detection
The construction of ScholarPhi

Major challenge: definition recognition

Solution: new algorithm for definition detection

Problems:
Difficult to specify what a definition is
Accuracy is still not high enough
.. we represent the **projected box** \( b_p \) as a 4-dimensional vector \( \mathbf{d}_{\{i,j\}} = [d_{\{t\{i,j\}\}} , d_{\{l\{i,j\}\}} , d_{\{b\{i,j\}\}} , d_{\{r\{i,j\}\}}] \), where \( d_t \), \( d_l \), \( d_b \), \( d_r \) are the distances between the current pixel location \((i,j)\) and the top, left, bottom, and right boundaries of \( b_p \), respectively. ..
.. we represent the projected box $b_p^l$ as a 4-dimensional vector $d_{t_{i,j}}^l, d_{l_{i,j}}^l, d_{b_{i,j}}^l, d_{r_{i,j}}^l$, where $d_{t^l}, d_{l^l}, d_{b^l}, d_{r^l}$ are the distances between the current pixel location $(i,j)$ and the top, left, bottom, and right boundaries of $b_p^l$, respectively. ..
Heddex/Taddex: State-of-the-art Definition Recognition

These models are seeing >70% F1 accuracy on definition prediction.

Kang et al., EMNLP-SDP '20
Can a LLM Solve These Problems?

GPT-4 is the leading LLM: How well does it do?
Why GPT-4?
Comparisons from the Llama 2 Paper

<table>
<thead>
<tr>
<th>Benchmark (shots)</th>
<th>GPT-3.5</th>
<th>GPT-4</th>
<th>PaLM</th>
<th>PaLM-2-L</th>
<th>Llama 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>MMLU (5-shot)</td>
<td>70.0</td>
<td>86.4</td>
<td>69.3</td>
<td>78.3</td>
<td>68.9</td>
</tr>
<tr>
<td>TriviaQA (1-shot)</td>
<td>–</td>
<td>–</td>
<td>81.4</td>
<td>86.1</td>
<td>85.0</td>
</tr>
<tr>
<td>Natural Questions (1-shot)</td>
<td>–</td>
<td>–</td>
<td>29.3</td>
<td>37.5</td>
<td>33.0</td>
</tr>
<tr>
<td>GSM8K (8-shot)</td>
<td>57.1</td>
<td>92.0</td>
<td>56.5</td>
<td>80.7</td>
<td>56.8</td>
</tr>
<tr>
<td>HumanEval (0-shot)</td>
<td>48.1</td>
<td>67.0</td>
<td>26.2</td>
<td>–</td>
<td>29.9</td>
</tr>
<tr>
<td>BIG-Bench Hard (3-shot)</td>
<td>–</td>
<td>–</td>
<td>52.3</td>
<td>65.7</td>
<td>51.2</td>
</tr>
</tbody>
</table>

GPT-4 is trained on text + images and can do a surprising wide range of tasks.

To answer question I.1.a, we need to express the temperature $T(x)$ in every point of the conductive bar, which is characterized by the abscissa $x$. We are given that $T(0) = T_0$ and $T(L) = T_b$. 
GPT-4 is training on text + images and can do a surprising wide range of tasks

I am going to give you a description of a short scene, and I would like you to provide me with a DOT graph representation in a code block where each person in the scene is a node, and whenever two characters interact there is an edge between them:

When first entering the office, Alice greeted Bob who was sitting at his desk and working at the final report. Until the end of the day, Alice never left her cubicle, only briefly talking with Claire during lunch. Most of the afternoon, Daniel and Bob were doing pair programming in the side room, and at the end of the day Dani, Claire and Ed had a meeting.
GPT-4 is training on text + images and can do a surprising wide range of tasks.

The above DOT graph represents the scene described in the prompt. Each character is a node, and whenever two characters interact, there is an edge between them. For example, Alice greets Bob when she first enters the office, so there is an edge between Alice and Bob. Similarly, Alice briefly talks with Claire during lunch, so there is an edge between Alice and Claire. Daniel and Bob do pair programming, so there is an edge between them. Finally, Daniel, Claire, and Ed have a meeting at the end of the day, so there are edges between Daniel and Claire, Daniel and Ed, and Claire and Ed.

Does GPT-4 Solve the Definition Recognition Problem?

Anna Martin-Boyle
Dongyeop Kang
Marti A. Hearst
For this sentence, what do the labels in the set refer to?

Each character in a sentence is assigned with a unique label from the set \{B, M, E, S\}, where \{B, M, E\} indicate Begin, Middle, End of a multi-character word respectively, and S denotes a Single character word.
For this sentence, what do the labels in the set refer to?

Each character in a sentence is assigned with a unique label from the set \{B, M, E, S\}, where \{B, M, E\} indicate Begin, Middle, End of a multi-character word respectively, and S denotes a Single character word.

In the given sentence, the labels from the set \{B, M, E, S\} are used to indicate the segmentation of words within the sentence. Each character in a word is assigned a label according to its position within the word. The labels have the following meanings:

- **B**: Begin - This label indicates the beginning character of a multi-character word.
- **M**: Middle - This label indicates a character that is in the middle of a multi-character word.
- **E**: End - This label indicates the end character of a multi-character word.
- **S**: Single - This label is used for single-character words.
For this sentence, what do the symbols refer to?
We represent the projected box $b_p^l$ as a 4-dimensional vector $\mathbf{d}_{(i,j)}^l = [d_{(t_{(i,j)}}^l, d_{(l_{(i,j)}}^l, d_{(b_{(i,j)}}^l, d_{(r_{(i,j)}}^l]$, where $d_{(t^l}, d_{l^l}, d_{b^l}, d_{r^l}$ are the distances between the current pixel location $(i, j)$ and the top, left, bottom, and right boundaries of $b_p^l$, respectively.
For this sentence, what do the symbols refer to?
We represent the projected box $b_p^l$ as a 4-dimensional vector $\mathbf{d}_{(i,j)}^l = [d_{(t_{(i,j)}}^l, d_{(l_{(i,j)}}^l, d_{(b_{(i,j)}}^l, d_{(r_{(i,j)}}^l]$, where $d_{t^l}$, $d_{l^l}$, $d_{b^l}$, and $d_{r^l}$ are the distances between the current pixel location $(i, j)$ and the top, left, bottom, and right boundaries of $b_p^l$, respectively. ..

In the given sentence, the symbols refer to various components and values related to representing a projected box. Here's a breakdown of what each symbol refers to:

- **$b_p^l$:** This refers to the projected box labeled as "$b_p^l". It's a box or rectangle that has been projected onto some context or space.
- **$\mathbf{d}_{(i,j)}^l$:** This represents a 4-dimensional vector labeled as "$\mathbf{d}_{(i,j)}^l". It's a vector used to store information about the projected box at the pixel location $(i,j)$.
- **$d_{(t_{(i,j)}}^l$:** This refers to the distance labeled as "$d_{(t_{(i,j)}}^l". It represents the distance between the current pixel location $(i, j)$ and the top boundary of the projected box.
- **$d_{(l_{(i,j)}}^l$:** This refers to the distance labeled as "$d_{(l_{(i,j)}}^l". It represents the distance between the current pixel location $(i, j)$ and the left boundary of the projected box.
- **$d_{(b_{(i,j)}}^l$:** This refers to the distance labeled as "$d_{(b_{(i,j)}}^l". It represents the distance between the current pixel location $(i, j)$ and the bottom boundary of the projected box.
- **$d_{(r_{(i,j)}}^l$:** This refers to the distance labeled as "$d_{(r_{(i,j)}}^l". It represents the distance between the current pixel location $(i, j)$ and the right boundary of the projected box.
Does GPT-4 Solve the Definition Recognition Problem?

It does well, but our specialized algorithm still performs (a bit) better.

<table>
<thead>
<tr>
<th></th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>TADDEX on terms</td>
<td>81.5</td>
</tr>
<tr>
<td>GPT-4 on terms</td>
<td>79.7</td>
</tr>
<tr>
<td>TADDEX on definitions</td>
<td>73.6</td>
</tr>
<tr>
<td>GPT-4 on definitions</td>
<td>70.4</td>
</tr>
</tbody>
</table>
Does GPT-4 Solve The Document Recognition Problem?

Can it recognize equation structure and location?

SHANNON ZEJIANG SHEN

I am a first year PhD Student at MIT CSAIL, working at the intersection between NLP and HCI.
VILA
Improved Structured Content Extraction from Scientific PDFs Using Visual Layout Groups
@ ACL TACL ’22.

LAYOUTPARSER
A Unified Toolkit for Deep Learning Based Document Image Analysis
@ ICDAR ’21.

SHANNON ZEJIANG SHEN
I am a first year PhD Student at MIT CSAIL, working at the intersection between NLP and HCI.
Layout Analysis – in 4 Lines of Code
Transform document image analysis pipelines with the full power of Deep Learning.

What is Layout Parser?
A Unified Toolkit for Deep Learning Based Document Image Analysis

Accurate Layout Detection with a Simple and Clean Interface
With the help of state-of-the-art deep learning models, Layout Parser enables extracting complicated document structures using only several lines of code. This method is also more robust and generalizable as no sophisticated rules are involved in this process.
A unified Toolkit for Processing, Representing, and Manipulating Visually-Rich Scientific Documents

(under review)
PaperMage allows for the composition of different machine learning and layout analysis models.

Access the various layers via python.
Crowdsourcing provides a scalable and efficient way to construct labeled datasets for training machine learning systems. However, creating comprehensive label guidelines for crowd-workers is often prohibitive even for seemingly simple concepts. Incomplete or ambiguous label guidelines can then result in differing interpretations of concepts and inconsistent labels. Existing approaches for improving label quality, such as worker screening or detection of poor work, are ineffective for this problem and can lead to rejection of honest work and a missed opportunity to capture rich interpretations about data. We introduce Revolt, a collaborative approach that brings ideas from expert annotation workflows to crowd-based labeling. Revolt eliminates the burden of creating detailed label guidelines by harnessing crowd disagreements to identify ambiguous concepts and create rich structures (groups of semantically related items) for post-hoc label decisions. Experiments comparing Revolt to traditional crowdsourced labeling show that Revolt produces high quality labels without requiring label guidelines in turn for an increase in monetary cost. This upfront cost, however, is mitigated by Revolt’s ability to produce reusable structures that can accommodate a variety of label boundaries without requiring new data to be collected. Further comparisons of Revolt’s collaborative and non-collaborative variants show that collaboration reaches higher label accuracy with lower monetary cost.

ACM Classification Keywords
H.5.m. Information Interfaces and Presentation (e.g. HCI): Miscellaneous

Author Keywords
crowdsourcing; machine learning; collaboration; real-time

INTRODUCTION
From conversational assistants on mobile devices, to facial
PaperMage supports structured algorithm comparison

<table>
<thead>
<tr>
<th>Structure Category</th>
<th>GROBID\textsubscript{CRF}</th>
<th>GROBID\textsubscript{NN}</th>
<th>RoBERTa</th>
<th>I-VILA</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>P</td>
<td>R</td>
<td>F1</td>
<td>P</td>
</tr>
<tr>
<td>Abstract</td>
<td>81.9</td>
<td>89.1</td>
<td>85.3</td>
<td>85.3</td>
</tr>
<tr>
<td>Author</td>
<td>55.2</td>
<td>42.6</td>
<td>48.1</td>
<td>75.1</td>
</tr>
<tr>
<td>Bibliography</td>
<td>96.5</td>
<td>98.6</td>
<td>97.5</td>
<td>95.5</td>
</tr>
<tr>
<td>Caption</td>
<td>70.3</td>
<td>70.0</td>
<td>70.2</td>
<td>70.2</td>
</tr>
<tr>
<td>Equation</td>
<td>71.1</td>
<td>85.3</td>
<td>77.6</td>
<td>71.1</td>
</tr>
<tr>
<td>Figure</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>Footer</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>Footnote</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>Header</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>Keywords</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>List</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>Paragraph</td>
<td>94.5</td>
<td>89.8</td>
<td>92.1</td>
<td>94.4</td>
</tr>
<tr>
<td>Section</td>
<td>83.0</td>
<td>79.4</td>
<td>81.1</td>
<td>83.0</td>
</tr>
<tr>
<td>Table</td>
<td>97.3</td>
<td>58.6</td>
<td>73.2</td>
<td>97.9</td>
</tr>
<tr>
<td>Title</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
</tbody>
</table>

| Macro Avg (Full S2-VL) | 40.6 | 38.3 | 39.1 | 42.0 | 36.5 | 37.6 | 75.9 | 80.0 | 76.8 | 92.0 | 94.1 | 92.7 |
| Macro Avg (Grobid Subset) | 81.2 | 76.7 | 78.9 | 84.1 | 73.0 | 78.2 | 82.6 | 83.9 | 83.2 | 92.2 | 95.2 | 93.7 |
Does GPT-4 Solve The Document Recognition Problem?

Can it recognize equation structure and location?
A typical PDF Parsing Pipeline

Excerpt from Attention is all you need by Vaswani et al., NeurIPS 2017
3.2.1 Scaled Dot-Product Attention

We call our particular attention “Scaled Dot-Product Attention” (Figure 2). The input consists of queries and keys of dimension $d_k$, and values of dimension $d_v$. We compute the dot products of the query with all keys, divide each by $\sqrt{d_k}$, and apply a softmax function to obtain the weights on the values.

In practice, we compute the attention function on a set of queries simultaneously, packed together into a matrix $Q$. The keys and values are also packed together into matrices $K$ and $V$. We compute the matrix of outputs as:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}V\right)$$ (1)

The two most commonly used attention functions are additive attention [2], and dot-product (multiplicative) attention. Dot-product attention is identical to our algorithm, except for the scaling factor of $\frac{1}{\sqrt{d_k}}$. Additive attention computes the compatibility function using a feed-forward network with a single hidden layer. While the two are similar in theoretical complexity, dot-product attention is much faster and more space-efficient in practice, since it can be implemented using highly optimized matrix multiplication code.

While for small values of $d_k$, the two mechanisms perform similarly, additive attention outperforms dot product attention without scaling for larger values of $d_k$ [3]. We suspect that for large values of $d_k$, the dot products grow large in magnitude, pushing the softmax function into regions where it has extremely small gradients \(^4\). To counteract this effect, we scale the dot products by $\frac{1}{\sqrt{d_k}}$.

3.2.2 Multi-Head Attention

The input consists of queries and keys of dimension $d_k$, and values of dimension $d_v$. We compute the dot products of the query with all keys, divide each by $\sqrt{d_k}$, and apply a softmax function to obtain the weights on the values.

In practice, we compute the attention function on a set of queries simultaneously, packed together into a matrix $Q$. The keys and values are also packed together into matrices $K$ and $V$. We compute the matrix of outputs as:

$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V$$ (1)

The two most commonly used attention functions are additive attention [2], and dot-product (multiplicative) attention. Dot-product attention is identical to our algorithm, except for the scaling factor of $\frac{1}{\sqrt{d_k}}$. Additive attention computes the compatibility function...
GPT-4 can analyze pdf text output ...

... on the values. In practice, we compute the attention function on a set of queries simultaneously, packed together into a matrix $Q$. The keys and values are also packed together into matrices $K$ and $V$. We compute the matrix of outputs as:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{dk}}\right)V$$

Can you detect the equations from the PDF text and convert it into latex?

But it is not correct.
Here is the original equation and the subscript is wrong:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

... but the lack of structure can cause errors in downstream tasks
3.2.1 Scaled Dot-Product Attention

We call our particular attention "Scaled Dot-Product Attention" (Figure 2). The input consists of queries and keys of dimension $d_k$, and values of dimension $d_v$. We compute the dot products of the query with all keys, divide each by $\sqrt{d_k}$, and apply a softmax function to obtain the weights on the values.

In practice, we compute the attention function on a set of queries simultaneously, packed together into an matrix $Q$. The keys and values are also packed together into matrices $K$ and $V$. We compute the matrix of outputs as:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{Q K^T}{\sqrt{d_k}}\right)V$$

The two most commonly used attention functions are additive attention [2] and dot product (multiplicative) attention. Dot product attention is identical to our algorithm, except for the scaling factor of $\frac{1}{\sqrt{d_k}}$. Additive attention computes the compatibility function using a feed-forward network with a single hidden layer. While the two are similar in theoretical complexity, dot product attention is much faster and more space-efficient in practice, since it can be implemented using highly optimized matrix multiplication code.

While for small values of $d_k$, the two mechanisms perform similarly, additive attention outperforms dot product attention without scaling for larger values of $d_k$. We suspect that for larger values of $d_k$, the dot products grow large in magnitude, pushing the softmax function into regions where it has extremely small gradients. To combat this effect, we scale the dot products by $\frac{1}{\sqrt{d_k}}$.

3.2.2 Multi-Head Attention

Instead of performing a single attention function with $d_{att}$-dimensional keys, values and queries, we found it beneficial to linearly project the queries, keys and values $h$ times with different, learned linear projections to $d_k$, $d_v$ and $d_q$, dimensions, respectively. On each of these projected versions of queries, keys and values we then perform the attention function in parallel, yielding $d_{att}$-dimensional output values. These are concatenated and once again projected, resulting in the final values, as depicted in Figure 2.
What is inside a PDF File?

PDF Code → PDF Objects
Describes how to display text and other objects on a page.

BT /F61 9.9626 Tf 1.02 0 0 1 109 509.981 Tm
[(Figure)–341(2:)–495((left))–340(Scaled)–341(Dot–
Product)–341(Attention.)–590((right))–341(Multi–
Head)–341(At)1(tention)–341(consists)–341(of)–340(
se)24(v)15(eral)]1 0 0 1 108 499.072 Tm
[(attention)–250(layers)–250(running)–250(in)–250(parallel.))]0 g 0 G
### Scaled Dot-Product Attention

We call our particular attention "Scaled Dot-Product Attention" (Figure 2). The input consists of queries and keys of dimension \(d_q\), and values of dimension \(d_v\). We compute the dot products of the query with all keys, divide each by \(\sqrt{d_k}\), and apply a softmax function to obtain the weights on the values.

In practice, we compute the attention function on a set of queries simultaneously, packed together into a matrix \(Q\). The keys and values are also packed together into matrices \(K\) and \(V\). We compute the matrix of outputs as:

\[
\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V
\]  

The two most commonly used attention functions are additive attention \([2]\), and dot-product (multiplicative) attention. Dot-product attention is identical to our algorithm, except for the scaling factor of \(\sqrt{d_k}\). Additive attention computes the compatibility function using a feed-forward network with a single hidden layer. While the two are similar in theoretical complexity, dot-product attention is much faster and more space-efficient in practice, since it can be implemented using highly optimized matrix multiplication code.
3.2.1 Scaled Dot-Product Attention

We call our particular attention "Scaled Dot-Product Attention" (Figure 2). The input consists of queries and keys of dimension $d_k$, and values of dimension $d_v$. We compute the dot products of the query with all keys, divide each by $\sqrt{d_k}$, and apply a softmax function to obtain the weights on the values.

In practice, we compute the attention function on a set of queries simultaneously, packed together into a matrix $Q$. The keys and values are also packed together into matrices $K$ and $V$. We compute the matrix of outputs as:

$$Attention(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V$$

(1)

The two most commonly used attention functions are additive attention [2], and dot-product (multiplicative) attention. Dot-product attention is identical to our algorithm, except for the scaling factor of $\sqrt{d_k}$. Additive attention computes the compatibility function using a feed-forward network with a single hidden layer. While the two are similar in theoretical complexity, dot-product attention is much faster and more space-efficient in practice, since it can be implemented using highly optimized matrix multiplication code.
3.2.1 Scaled Dot-Product Attention

We call our particular attention “Scaled Dot-Product Attention” (Figure 2). The input consists of queries and keys of dimension \( d_q \) and values of dimension \( d_v \). We compute the dot products of the query with all keys, divide each by \( \sqrt{d_k} \), and apply a softmax function to obtain the weights on the values.

In practice, we compute the attention function on a set of queries simultaneously, packed together into a matrix \( Q \). The keys and values are also packed together into matrices \( K \) and \( V \). We compute the matrix of outputs as:

\[
\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}V\right)
\]

The two most commonly used attention functions are additive attention \([2]\), and dot-product (multiplicative) attention. Dot-product attention is identical to our algorithm, except for the scaling factor of \( \frac{1}{\sqrt{d_k}} \). Additive attention computes the compatibility function using a feed-forward network with a single hidden layer. While the two are similar in theoretical complexity, dot-product attention is much faster and more space-efficient in practice, since it can be implemented using highly optimized matrix multiplication code.
It turns out that GPT-4 can analyze PDF code!

It gets the subscript correct that it missed from pdf text alone.
How does this work? Let’s dive deeper into the PDF code

There is an inline equation in this pdf code:

1. \( \gamma \) prints the letter \( \gamma \) with a regular font size.
2. 1 prints the number 1 with a smaller font size, and the text is shifted downside. Thus 1 is a subscript. The code is `'_{1}`.

Therefore, the equation LaTeX is `$\gamma_{_{1}}=1$`.

Correct Parsing

Original PDF

\[
\Pr[M(D) \in S] \leq \exp(\epsilon) \Pr[M(D') \in S] + \delta.
\]

PDF Code Input (3800 chars)

\[
\ldots/F24 9.9626 \ Tf -147.876 -18.267 \ Td [(\Pr[])/F27 9.9626 \ Tf 13.449 0 \ Td [(M)]/F24 9.9626 \ Tdf 10.752 0 \ Td [(())]/F27 9.9626 \ Tf 3.874 0 \ Td [(D)]/F24 9.9626 \ Tf 8.525 0 \ Td [(())]/F30 9.9626 \ Tf 6.642 0 \ Td [(\epsilon)]/F27 9.9626 \ Tf 9.409 0 \ Td [(S)]/F24 9.9626 \ Tf 6.683 0 \ Td [(())]/F30\ldots
\]

\[
\rightarrow \text{GPT-4} \rightarrow \text{Latex:}
\]

\[
\Pr[M(D) \in S] \leq \exp(\epsilon) \Pr[M(D') \in S] + \delta.
\]

PDF Text Input (717 chars)

\[
\ldots\text{datasets } D, D' \text{ and for any } S \subseteq \text{range}(M), \quad 
\Pr[M(D) \in S] \leq \exp(\epsilon) \Pr[M(D') \in S] + \delta.
\]

We say two datasets $D, D' \in X$ are neighboring if they differ on at most an individual’s participation. Two additional steps are added to the FL algorithm to ensure a DP guar-...
Failure Cases
Complex Subscript Structure

Original PDF
\[ \mathcal{E}(f_{\text{pop}}^*) \leq \tilde{O}\left(\frac{\alpha}{\rho^2} \right). \]

PDF Code → GPT-4 → Latex
\[ E(f_{\text{pop}}^*) \leq \tilde{O}\left(\frac{\alpha}{\rho^2} \right). \]

PDF Text → GPT-4 → Latex
\[ E(f_{\text{pop}}^*) \leq \tilde{O}\left(\frac{\alpha}{\rho^2} \right). \]
Failure Cases
Complex and Long Equations

Original PDF
\[ \frac{1}{\lambda} \cdot \frac{C - Npq}{N(1 - p) + 1} \leq \mathbb{E}[U_k] \leq \frac{C}{\lambda(N - C)}. \]

PDF Code → GPT-4 → Latex
\[ \lambda \cdot C - Npq \quad \text{①} \quad N(1 - p) + 1 \leq \mathbb{E}[U_k] \leq \frac{C}{\lambda(N - C)} \quad \text{③} \]

PDF Text → GPT-4 → Latex
\[ \frac{1}{\lambda \cdot C} = \frac{Npq}{N(1 - p) + 1} \leq \mathbb{E}[U_k] \leq \frac{C}{\lambda(N - C)}. \]

In-context learning examples

We manually write the prompts to teach GPT-4 how to parse the equations from the PDF Code.
Does GPT-4 Solve The Document Recognition Problem?

Stay tuned … we are working on getting detailed results

SHANNON ZEJIANG SHEN

I am a first year PhD Student at MIT CSAIL working at the intersection between NLP and HCI.
To consider: In light of the capabilities of LLMs, what is the role of classic representations?

For NLP: do we need syntax and parse trees?  
For document analysis: do we need layout structure?

Questions drawn from Kyle Lo & Dan Klein
In Summary

NLP + HCI + Document Analysis can Improve the Understandability of Scientific Papers

Scientific paper interactivity offers great research opportunities for the ICDAR community!
Bringing Scientific Papers to Life
Thank you!

Marti Hearst
UC Berkeley

In collaboration with the Allen Institute of AI