



Named Entity Recognition Using BERT and ELMo

Group 8 :
Mikaela Guerrero
Vikash Kumar
Nitya Sampath
Saumya Shah



Introduction to Named Entity Recognition

US GPE unveils world's most powerful supercomputer, beats China GPE . The US GPE has unveiled the world's most powerful supercomputer called 'Summit', beating the previous record-holder China GPE 's Sunway TaihuLight ORG . With a peak performance of 200,000 CARDINAL trillion calculations per second ORDINAL , it is over twice as fast as Sunway TaihuLight ORG , which is capable of 93,000 CARDINAL trillion calculations per second. Summit has 4,608 CARDINAL servers, which reportedly take up the size of two CARDINAL tennis courts.

Named entity recognition (NER) seeks to locate and classify named entities in text into predefined categories such as the names of persons, organizations, locations, expressions of times, quantities, monetary values, percentages, etc.

The goal of NER is to tag a set of words in a sequence with a label representing the kind of entity the word belongs to.

Named Entity Recognition is probably the first step in Information Extraction and it plays a key role in extracting structured information from documents and conversational agents.

NER in action

In fact, the two major components of a Conversational bot's NLU are Intent Classification and Entity Extraction. Each word of the sentence is labeled using the IOB scheme (Inside-Outside-Beginning) with an additional connection label to label words used to connect different named entities. These labels are then used to extract entities from our command

Play	Bohemian	Rhapsody	by	Queen
O	B-Song	I-Song	Connect	B-Artist

Every NER algorithm proceeds as a sequence of the following steps -

1. Chunking and text representation - eg. New York represents one chunk
2. Inference and ambiguity resolution algorithms - eg. Washington can be a name or a location
3. Modeling of Non-Local dependencies - eg. Garrett, garrett, and GARRETT should all be identified as the same entity
4. Implementation of external knowledge resources

Transfer learning and why is it relevant



After supervised learning – Transfer Learning will be the next driver of ML commercial success - Andrew NG

Humans have an inherent ability to transfer knowledge across tasks. What we acquire as knowledge while learning about one task, we utilize in the same way to solve related tasks. The more related the tasks, the easier it is for us to transfer, or cross-utilize our knowledge. For example - know math and statistics ☐ Learn machine learning

In the above scenario, we don't learn everything from scratch when we attempt to learn new aspects or topics. We transfer and leverage our knowledge from what we have learnt in the past.

Thus, the key motivation, especially considering the context of deep learning is the fact that most models which solve complex problems need a whole lot of data, and getting vast amounts of labeled data for supervised models can be really difficult, considering the time and effort it takes to label data points.

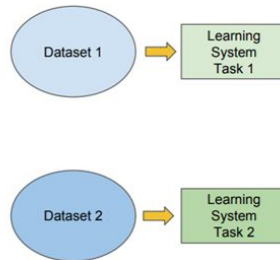
The Age of Transfer Learning

Transfer learning is a machine learning method where a model developed for a task is reused as the starting point for a model on a second task.

Conventional machine learning and deep learning algorithms, so far, have been traditionally designed to work in isolation. These algorithms are trained to solve specific tasks. The models have to be rebuilt from scratch once the feature-space distribution changes. Transfer learning is the idea of overcoming the isolated learning paradigm and utilizing knowledge acquired for one task to solve related ones.

Traditional ML

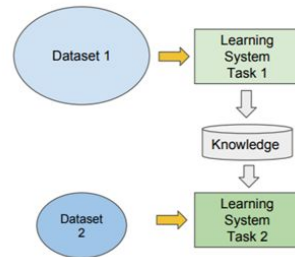
- Isolated, single task learning:
 - Knowledge is not retained or accumulated. Learning is performed w.o. considering past learned knowledge in other tasks



vs

Transfer Learning

- Learning of a new task relies on the previous learned tasks:
 - Learning process can be faster, more accurate and/or need less training data



Overview of the presentation



The original state of the art in Named Entity Recognition

The paper proposed by Lample et al. (2016) - Neural Architectures for Named Entity Recognition became the state-of-the-art in NER

However it did not employ any transfer learning techniques.

Discuss the influence of transfer learning to NER

With the other papers, we see the influence of transfer learning and especially language models in NER.

Implementation of our project

We talk about our proposed hypothesis and analysis methods.

Progression of NER systems from no incorporation of language models to language model based implementation.





Neural Architectures for Named Entity Recognition

Guillaume Lample♠ Miguel Ballesteros♣♠

Sandeep Subramanian♠ Kazuya Kawakami♠ Chris Dyer♠

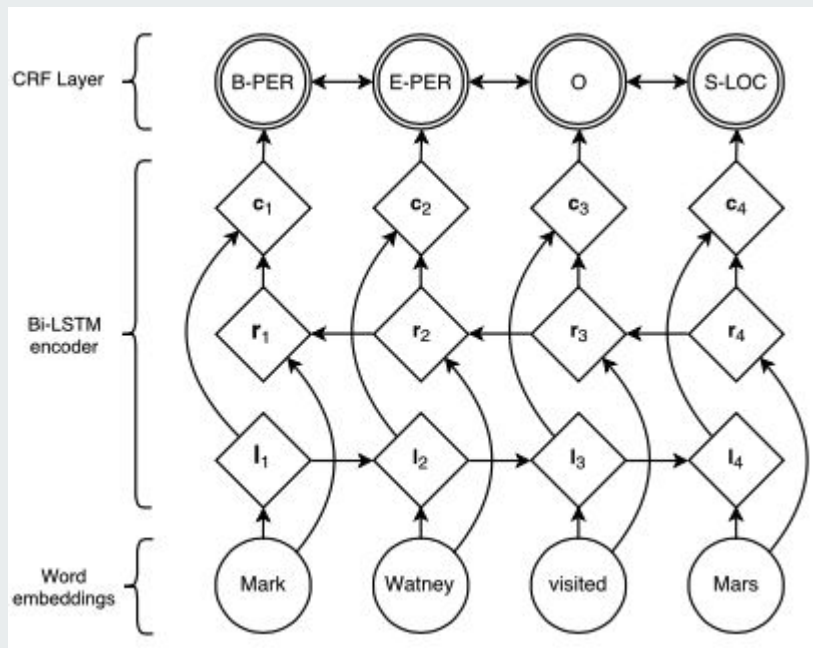
♠Carnegie Mellon University ♣NLP Group, Pompeu Fabra University

{glample, sandeeps, kkawakam, cdyer}@cs.cmu.edu,
miguel.ballesteros@upf.edu

Proposed by Lample et. al (2016), this was the first work on NER to completely drop hand-crafted features, i.e., they use no language-specific resources or features, just embeddings.

Lample, Guillaume, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. "Neural architectures for named entity recognition." arXiv preprint arXiv:1603.01360 (2016).

State-of-the-art for NER




























- The word embeddings are the concatenation of two vectors,
 - a vector made of character embeddings using two LSTMs
 - and a vector corresponding to word embeddings trained on external data.
- The rationale behind this idea is that many languages have orthographic or morphological evidence that a word or sequence of words is a named-entity or not, so they use character-level embeddings to try to capture these evidences.
- The embeddings for each word in a sentence are then passed through a forward and backward LSTM, and the output for each word is then fed into a CRF layer.

Examples of how using language models has helped accuracy scores of Named Entity Recognition

Leaderboards

[Add a Result](#)

TREND	DATASET	BEST METHOD	PAPER TITLE	PAPER	CODE	COMPARE
	CoNLL 2003 (English)	 CNN Large + fine-tune	Cloze-driven Pretraining of Self-attention Networks			See all
	Ontonotes v5 (English)	 BERT-MRC+DSC	Dice Loss for Data-imbalanced NLP Tasks			See all
	ACE 2005	 seq2seq+BERT+Flair	Neural Architectures for Nested NER through Linearization			See all
	GENIA	 seq2seq+BERT+Flair	Neural Architectures for Nested NER through Linearization			See all
	JNLPBA	 CollaboNet	CollaboNet: collaboration of deep neural networks for biomedical named entity recognition			See all
	BC5CDR	 NER+PA+RL (PubMed)	Reinforcement-based denoising of distantly supervised NER with partial annotation			See all
	SciERC	 SpERT	Span-based Joint Entity and Relation Extraction with Transformer Pre-training			See all

Transfer Learning Using Pre-trained Language Models





Neural Architectures for Nested NER through Linearization

Jana Straková and **Milan Straka** and **Jan Hajič**

Charles University

Faculty of Mathematics and Physics

Institute of Formal and Applied Linguistics

`{strakova, straka, hajic}@ufal.mff.cuni.cz`

Overview

Task:

- Nested Named Entity Recognition (NER)
- Flat NER

Architectures:

- LSTM-CRF
- seq2seq

Contextual Embeddings:

- ELMo
- BERT
- Flair

Datasets:

- ACE-2004 & 2005 (English)
- GENIA (English)
- CNEC (Czech)
- CoNLL-2002 (Dutch & Spanish)
- CoNLL-2003 (English & German)



Methodology (Data)



Nested NE BILOU Encoding:

in	O
the	B-ORG
US	I-ORG U-GPE
Federal	I-ORG
District	I-ORG U-GPE
Court	I-ORG
of	I-ORG
New	I-ORG B-GPE
Mexico	L-ORG L-GPE
.	O

Datasets:

- **Nested NE Corpora:**
ACE-2004, ACE-2005, GENIA, CNEC
- **Corpora used to evaluate Flat NER:**
CoNLL-2002 (Dutch & Spanish), CoNLL-2003 (English & German)

Split:

- Train portion used for training
- Development portion used for hyperparameter tuning
- Models trained on concatenated train+dev portions
- Models evaluated on test portion

Methodology (Models)



1) LSTM-CRF

- **Encoder:** bi-directional LSTM
- **Decoder:** CRF

2) Sequence-to-sequence (seq2seq)

- **Encoder:** bi-directional LSTM
- **Decoder:** LSTM
- Hard attention on words whose label(s) is being predicted

Architecture Details:

- Lazy Adam optimizer with $\beta_1 = 0.9$ and $\beta_2 = 0.98$
- Mini-batches of size 8
- Dropout with rate 0.5

Baseline Model Embeddings:

- pretrained (using word2Vec and FastText)
- end-to-end (input forms, lemmas, POS tags)
- character-level (using bidirectional GRUs)

Contextual Word Embeddings:

- ELMo (for English)
- BERT (for all languages)
- Flair (for all languages except Spanish)

Results

- seq2seq appears to be suitable for more complex/nested corpora
- LSTM-CRF simplicity is good for flat corpora with shorter and less overlapping entities
- Adding contextual embeddings beats previous literature in all cases aside from CoNLL-2003 German

Nested NER results (F1)

model	ACE-2004	ACE-2005	GENIA	CNEC 1.0
(Finkel and Manning, 2009)**	—	—	70.3	—
(Lu and Roth, 2015)**	62.8	62.5	70.3	—
(Muis and Lu, 2017)**	64.5	63.1	70.8	—
(Katiyar and Cardie, 2018)	72.70	70.5	73.6	—
(Ju et al., 2018)*	—	72.2	74.7	—
(Wang and Lu, 2018)	75.1	74.5	75.1	—
(Straková et al., 2016)	—	—	—	81.20
LSTM-CRF	72.26	71.62	76.23	80.28
LSTM-CRF+ELMo	78.72	78.36	75.94	—
LSTM-CRF+BERT	81.48	79.95	77.80	85.67
LSTM-CRF+Flair	77.65	77.25	76.65	81.74
LSTM-CRF+BERT+ELMo	80.07	80.04	76.29	—
LSTM-CRF+BERT+Flair	81.22	80.82	77.91	85.70
LSTM-CRF+ELMo+BERT+Flair	80.19	79.85	76.56	—
seq2seq	77.08	75.36	76.44	82.96
seq2seq+ELMo	81.94	81.95	77.33	—
seq2seq+BERT	84.33	83.42	78.20	86.73
seq2seq+Flair	81.38	79.83	76.63	83.55
seq2seq+BERT+ELMo	84.32	82.15	77.77	—
seq2seq+BERT+Flair	84.40	84.33	78.31	86.88
seq2seq+ELMo+BERT+Flair	84.07	83.41	78.01	—

Flat NER results (F1)

model	English	German	Dutch	Spanish
(Gillick et al., 2016)	86.50	76.22	82.84	82.95
(Lample et al., 2016)	90.94	78.76	81.74	85.75
ELMo (Peters et al., 2018)	92.22	—	—	—
Flair (Akbi et al., 2018)	93.09	88.32	—	—
BERT (Devlin et al., 2018)	92.80	—	—	—
LSTM-CRF	90.72	79.89	87.42	86.34
LSTM-CRF+ELMo	92.58	—	—	—
LSTM-CRF+BERT	92.94	84.53	92.48	88.77
LSTM-CRF+Flair	92.25	82.35	88.31	—
LSTM-CRF+BERT+ELMo	92.93	—	—	—
LSTM-CRF+BERT+Flair	93.22	84.44	92.69	—
LSTM-CRF+ELMo+BERT+Flair	93.38	—	—	—
seq2seq	90.77	79.09	87.59	86.04
seq2seq+ELMo	92.43	—	—	—
seq2seq+BERT	92.98	84.19	92.46	88.81
seq2seq+Flair	91.87	82.68	88.67	—
seq2seq+BERT+ELMo	92.99	—	—	—
seq2seq+BERT+Flair	93.00	85.10	92.34	—
seq2seq+ELMo+BERT+Flair	93.07	—	—	—

Conclusion



- Written during advent of using pre-trained language models for Transfer Learning
- Examined the differing strengths of two standard architectures (LSTM-CRF & seq2seq) for NER
- Surpassed state-of-the-art results for NER using contextual word embeddings



Transfer Learning in Biomedical Natural Language Processing: An Evaluation of BERT and ELMo on Ten Benchmarking Datasets

Yifan Peng Shankai Yan Zhiyong Lu

National Center for Biotechnology Information

National Library of Medicine, National Institutes of Health

Bethesda, MD, USA

`{yifan.peng, shankai.yan, zhiyong.lu}@nih.gov`

Overview



Introducing the BLUE (Biomedical Language Understanding Evaluation) benchmark

5 tasks, 10 datasets:

Sentence Similarity

- BIOSSES
- MedSTS

Named Entity Recognition

- BC5CDR-disease
- BC5CDR-chemical
- ShARe/CLEF

Relation Extraction

- DDI
- ChemProt
- i2b2 2010

Document Multilabel Classification

- HoC

Inference Task

- MedNLI

Ran experiments using BERT and ELMo as two baseline models to better understand BLUE

Methodology - BERT



Training

- Pre-trained on PubMed abstracts and MIMIC-III clinical notes
- 4 models:
 - BERT-Base (P)*
 - BERT-Large (P)
 - BERT-Base (P+M)**
 - BERT-Large (P+M)
- (P) models were trained on PubMed abstracts only
- (P+M) models were trained on both PubMed abstracts and MIMIC clinical notes

Fine-tuning

- Sentence similarity
 - Pairs of sentences were combined into a single sentence
- Named entity recognition
 - BIO tagging
- Relation extraction
 - certain pairs of related named entities were replaced with predefined tags
 - “Citalopram protected against the RTI-76-induced inhibition of SERT binding”
 - “@CHEMICAL\$ protected against the RTI-76-induced inhibition of @GENE\$ binding”

Methodology - ELMo



Training

- Pre-trained on PubMed abstracts

Fine-tuning

- Similar strategies as with BERT
- Sentence extraction
 - Transformed the sequences of word embeddings into sentence embeddings
- Named-entity recognition
 - Concatenated GloVe embeddings, character embeddings and ELMo embeddings of each token
 - Fed them to a Bi-LSTM-CRF implementation for sequence tagging

Results

Task	Metrics	SOTA*	ELMo	BioBERT	Our BERT			
					Base (P)	Base (P+M)	Large (P)	Large (P+M)
MedSTS	Pearson	83.6	68.6	84.5	84.5	84.8	84.6	83.2
BIOSSES	Pearson	84.8	60.2	82.7	89.3	91.6	86.3	75.1
BC5CDR-disease	F	84.1	83.9	85.9	86.6	85.4	82.9	83.8
BC5CDR-chemical	F	93.3	91.5	93.0	93.5	92.4	91.7	91.1
ShARe/CLEFE	F	70.0	75.6	72.8	75.4	77.1	72.7	74.4
DDI	F	72.9	78.9	78.8	78.1	79.4	79.9	76.3
ChemProt	F	64.1	66.6	71.3	72.5	69.2	74.4	65.1
i2b2	F	73.7	71.2	72.2	74.4	76.4	73.3	73.9
HoC	F	81.5	80.0	82.9	85.3	83.1	87.3	85.3
MedNLI	acc	73.5	71.4	80.5	82.2	84.0	81.5	83.8
Total			78.8	80.5	82.2	82.3	81.5	79.2

Performance of various models on BLUE benchmark tasks

Conclusion






- BERT-Base trained on both PubMed abstracts and MIMIC-III notes performed best across all tasks
- BERT-Base (P+M) also outperforms state-of-the-art models in most tasks
- In named-entity recognition, BERT-Base (P) had the best performance

Introduction



BioBERT: a pre-trained biomedical language representation model for biomedical text mining

Jinhyuk Lee ^{1,†}, Wonjin Yoon ^{1,†}, Sungdong Kim ², Donghyeon Kim ¹,
Sunkyu Kim ¹, Chan Ho So ³ and Jaewoo Kang ^{1,3,*}

¹Department of Computer Science and Engineering, Korea University, Seoul 02841, Korea, ²Clova AI Research, Naver Corp, Seong-Nam 13561, Korea and ³Interdisciplinary Graduate Program in Bioinformatics, Korea University, Seoul 02841, Korea

*To whom correspondence should be addressed.

[†]The authors wish it to be known that the first two authors contributed equally.

Associate Editor: Jonathan Wren

Received on May 16, 2019; revised on July 29, 2019; editorial decision on August 25, 2019; accepted on September 5, 2019

Overview



BioBERT is a domain specific language representation model pre-trained on large scale biomedical corpora. Directly applying the advancements in NLP to biomedical text mining often yields unsatisfactory results due to a word distribution shift from general domain corpora to biomedical corpora

Tasks:

- Pre-train the BioBERT model
- Fine tune BioBERT on popular medical NLP tasks like NER, Relationship extraction(RE) and Question-Answering

Datasets:

- Training: PubMed Abstracts(4.5B words), PMC(13.5B words)
- Evaluation: NCBI Disease (Dogan et al., 2014, 2010 i2b2/VA (Uzuner et al., 2011), BC5CDR (Li et al., 2016), BC4CHEMD (Krallinger et al., 2015), Species-800 (Pafilis et al., 2013), BioASQ

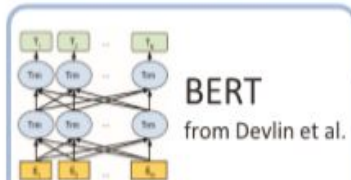
Illustration

Pre-training of BioBERT

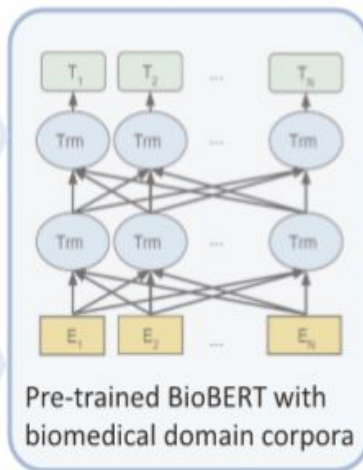
Pre-training Corpora

PubMed 4.5B words
PMC 13.5B words

Weight Initialization



BioBERT Pre-training



Fine-tuning of BioBERT

Task-Specific Datasets

Named Entity Recognition
NCBI disease, BC2GM, ...

Relation Extraction
EU-ADR, ChemProt, ...

Question Answering
BioASQ 5b, BioASQ 6b, ...

BioBERT Fine-tuning

the adult renal failure cause ...
▶ O O B I O ...

Variants in the @GENE\$ region
contribute to @DISEASE\$ susceptibility.
▶ True

What does mTOR stands for?
▶ mammalian target of rapamycin

Approach



- BioBERT uses the Word-Piece tokenization like BERT to handle OOV issues (medical domain terms are usually not found in colloquial English)
- For computational efficiency, whenever the Wiki + Books corpora were used, the weights were initialized with the pre-trained BERT Base model
- Hardware:
 - 8 NVIDIA V100 (32 GB) GPUs for pre-training. Training time was 23 days for BioBERT v1.1!
 - BERT was trained in 3.3 days on four DGX-2H nodes (a total of 64 Volta GPUs)
 - Single NVIDIA Titan Xp (12GB) GPU for fine-tuning on each task
 - Fine tuning is computationally simpler, with training time was less than 1 hour
 - 20 epochs to reach highest performance on NER dataset

Results

- Domain specific language models like BioBERT seem to perform better than generic purpose BERT

Type	Datasets	Metrics	SOTA	BERT	BioBERT v1.0			BioBERT v1.1
				(Wiki + Books)	(+ PubMed)	(+ PMC)	(+ PubMed + PMC)	(+ PubMed)
Disease	NCBI disease	P	<u>88.30</u>	84.12	86.76	86.16	89.04	88.22
		R	89.00	87.19	88.02	89.48	<u>89.69</u>	91.25
		F	88.60	85.63	87.38	87.79	<u>89.36</u>	89.71
	2010 i2b2/VA	P	<u>87.44</u>	84.04	85.37	85.55	87.50	86.93
		R	<u>86.25</u>	84.08	85.64	85.72	85.44	86.53
		F	<u>86.84</u>	84.06	85.51	85.64	86.46	<u>86.73</u>
	BC5CDR	P	89.61	81.97	85.80	84.67	85.86	<u>86.47</u>
		R	83.09	82.48	86.60	85.87	<u>87.27</u>	87.84
		F	<u>86.23</u>	82.41	86.20	85.27	86.56	87.15
Drug/chem.	BC5CDR	P	94.26	90.94	92.52	92.46	93.27	<u>93.68</u>
		R	92.38	91.38	92.76	92.63	93.61	<u>93.26</u>
		F	93.31	91.16	92.64	92.54	<u>93.44</u>	93.47
	BC4CHEMD	P	<u>92.29</u>	91.19	91.77	91.65	92.23	92.80
		R	90.01	88.92	<u>90.77</u>	90.30	90.61	91.92
		F	91.14	90.04	91.26	90.97	<u>91.41</u>	92.36
	BC2GM	P	81.81	81.17	81.72	82.86	85.16	<u>84.32</u>
		R	81.57	82.42	83.38	<u>84.21</u>	83.65	85.12
		F	81.69	81.79	82.54	83.53	<u>84.40</u>	84.72
Gene/protein	JNLPBA	P	74.43	69.57	71.11	71.17	<u>72.68</u>	72.24
		R	<u>83.22</u>	81.20	83.11	82.76	83.21	83.56
		F	<u>78.58</u>	74.94	76.65	76.53	<u>77.59</u>	77.49
	LINNAEUS	P	<u>92.80</u>	91.17	91.83	91.62	93.84	90.77
		R	94.29	84.30	84.72	85.48	<u>86.11</u>	85.83
		F	93.54	87.60	88.13	88.45	<u>89.81</u>	88.24
	Species-800	P	74.34	69.35	70.60	71.54	<u>72.84</u>	72.80
		R	<u>75.96</u>	74.05	75.75	74.71	<u>77.97</u>	75.36
		F	<u>74.98</u>	71.63	73.08	73.09	75.31	74.06

Domain specific NER

Table 9. Prediction samples from BERT and BioBERT on NER and QA datasets

Task	Dataset	Model	Sample
NER	NCBI disease	BERT	WT1 missense mutations, associated with male pseudohermaphroditism in Denys–Drash syndrome , fail to ...
		BioBERT	WT1 missense mutations, associated with male pseudohermaphroditism in Denys–Drash syndrome , fail to ...
	BC5CDR (Drug/Chem.)	BERT	... a case of oral penicillin anaphylaxis is described, and the terminology ...
		BioBERT	... a case of oral penicillin anaphylaxis is described, and the terminology ...
	BC2GM	BERT	Like the DMA , but unlike all other mammalian class II A genes, the zebrafish gene codes for two cysteine residues ...
		BioBERT	Like the DMA , but unlike all other mammalian class II A genes, the zebrafish gene codes for two cysteine residues ...
QA	BioASQ 6b-factoid		Q: Which type of urinary incontinence is diagnosed with the Q tip test?
		BERT	A total of 25 women affected by clinical stress urinary incontinence (SUI) were enrolled. After undergoing (...) Q-tip test, ...
		BioBERT	A total of 25 women affected by clinical stress urinary incontinence (SUI) were enrolled. After undergoing (...) Q-tip test, ...
			Q: Which bacteria causes erythrasma?
		BERT	Corynebacterium minutissimum is the bacteria that leads to cutaneous eruptions of erythrasma ...
		BioBERT	Corynebacterium minutissimum is the bacteria that leads to cutaneous eruptions of erythrasma ...

Conclusions



- BioBERT obtains higher F1 scores in biomedical NER (0.62% improvement over SOTA)
- BioBERT can recognize biomedical named entities that BERT cannot and can find the exact boundaries of named entities (although no accuracy scores are presented in the paper)
- Pre-training on domain specific tasks is essential to achieve better results
- Minimal task-specific architectural modifications required to build domain specific language models

Our project



We propose to analyze the use of language models for the task of Named Entity Recognition. This analysis ties in to the concept of transfer learning and for this project, we will examine how language models like BERT and ELMo learn named entities when trained on a specific task.

This analysis also extends from general Named Entity Recognition to domain-specific NER. We do our experiments on two datasets, the general NER dataset from CoNLL and the Movie Dataset from MIT.

Specifically, when a language model is trained on named entities, which layer identifies a named entity, which layers produce the associations with named entities and how a language representation model can understand word associations.

Proposed Implementation



- We will convert the problem into a sequence labeling task where the objective is to learn the IOB tags for the tokens. We will be using the “bert-base-cased” variant of BERT as it is more suited for the NER task.
- We will be using the AllenNLP framework to run our experiments which will allow us to track our runs by adjusting the configurations and ensuring reproducibility of the results.
- We intend to run our experiments on two datasets
 - A general dataset - the CoNLL dataset
 - A domain specific dataset - Movie dataset from MIT
- Our test set will be a list of sentences with manually annotated IOB tags and we will be comparing the f1 scores from the two models as our comparison metric.
- We wish to contrast how BERT and ELMo are trained on the task and the kind of scores they produce at the time of training on a general as well domain-specific NER.

AllenNLP Framework



- The AllenNLP framework allows us to treat each step in our algorithm as a black box
- With minimal changes to the main code we can pick and choose how we want to implement a particular task. For example - with few changes, we can use word embeddings from BERT or ELMo or GloVe
- The framework is almost like a black box - we specify the input, some config settings and the algorithm and the framework takes care of the implementation details
- We can also run several experiments on our project - for example compare NER with a CRF as the final layer versus a LSTM or an HMM etc
- It also allows us to customize the pipeline which bodes well for domain specific learning as well



Questions?

