A Review of
Named Entity Recognition

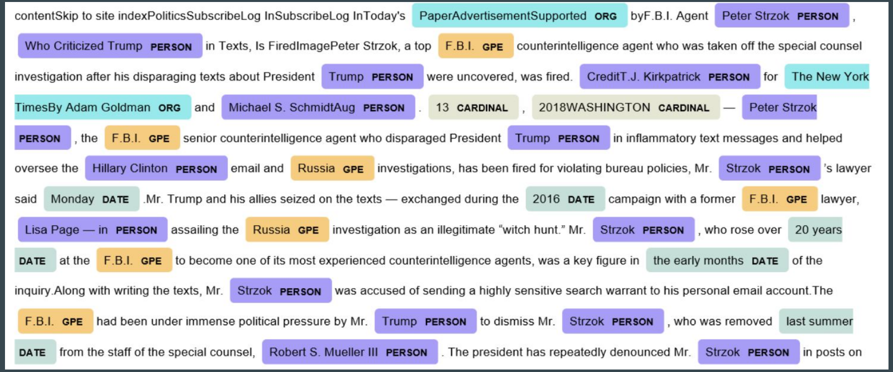
...

Table of Contents

What is Named Entity Recognition (NER) ?

NER, short for, Named Entity Recognition, is a standard Natural Language Processing problem which deals with information extraction. The primary objective is to locate and classify named entities in text into predefined categories such as the names of persons, organizations, locations, events, expressions of times, quantities, monetary values, percentages, etc.

To put it simply, NER deals with extracting the real-world entity from the text such as a person, an organization, or an event. Named Entity Recognition is also simply known as entity identification, entity chunking, and entity extraction. They are quite similar to POS(part-of-speech) tags.



Common Entity Types Identified by NER

Type	Description	Type	Description
Person	People	LAW	Named docs made into laws
NORP	Nationalities, Religious/Political Groups	LANGUAGE	Named Languages
FAC	Buildings, Airports, Highways, Bridges	DATE	Absolute/Relative Dates or Periods
ORG	Companies, Agencies, Institutions	PERCENT	% measures
GPE	Countries, Cities, States	QUANTITY	Measurements of Weights/ Distance
LOC	Non-GPE Locations	ORDINAL	First, second etc
Product	Objects, Vehicles, Food	MONEY	Monetary Values
EVENT	Named Hurricanes, Battles, Sport Events etc	TIME	Time smaller than a day
WORK_OF_ART	Titles of Books. Songs etc		

A typical Pipeline

NER is typically a **supervised** task. It needs annotated training data. Features are defined. From the annotated data, the ML model learns how to map these features to the entities.

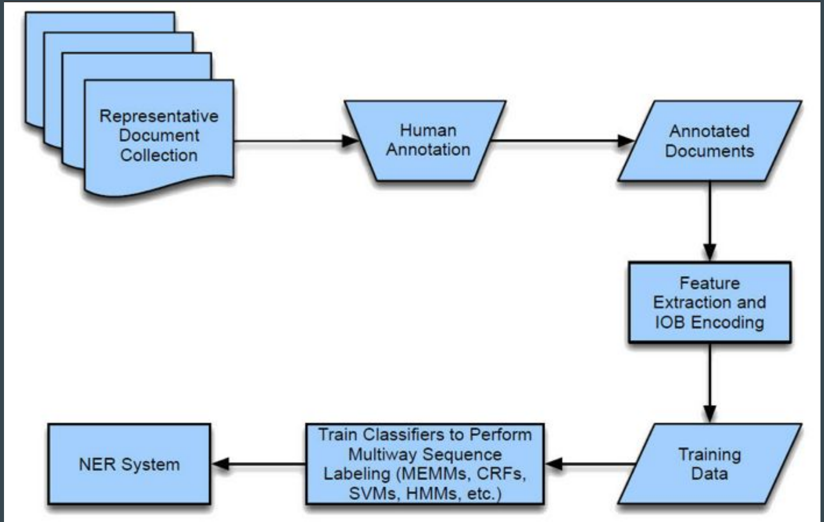
NER can be seen as a sequence labelling problem since it identifies a span of tokens and classifies it as a named entity.

NER can be solved in a manner similar to POS tagging or syntactic phrase chunking. Statistical models such as HMM, MEMM or CRF can be used.

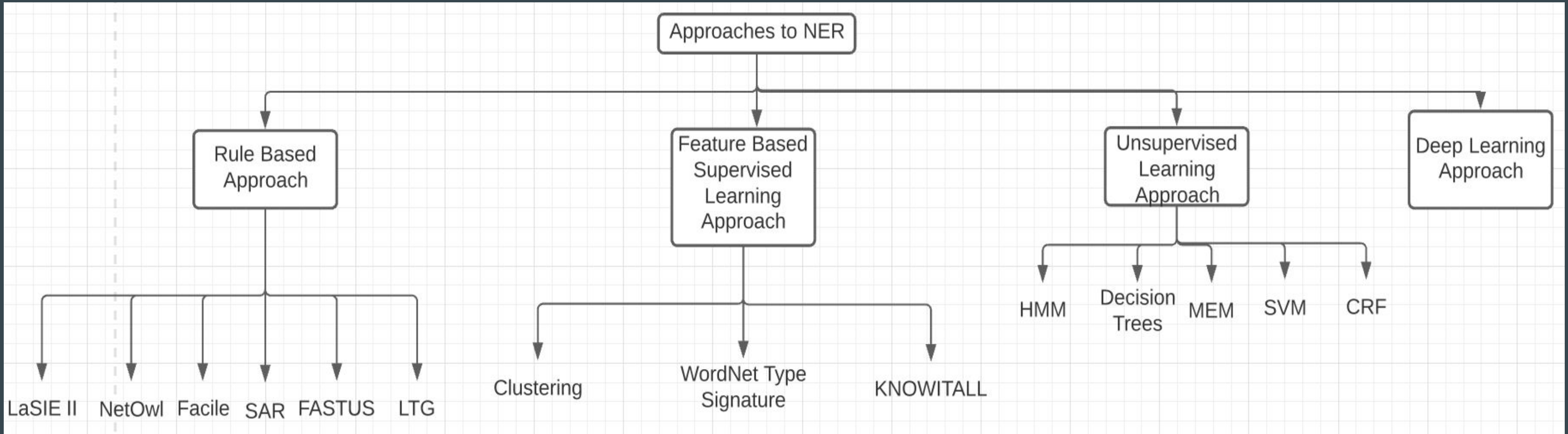
Annotating data manually is slow. A semi-automated approach is to give a file containing a set of patterns or rules. Another approach is to start with an existing model and manually correct mistakes.[1]

A practical approach is for a model to label unambiguous entities in the first pass. Subsequent passes make use of already labelled entities to resolve ambiguities about other entities.[2]

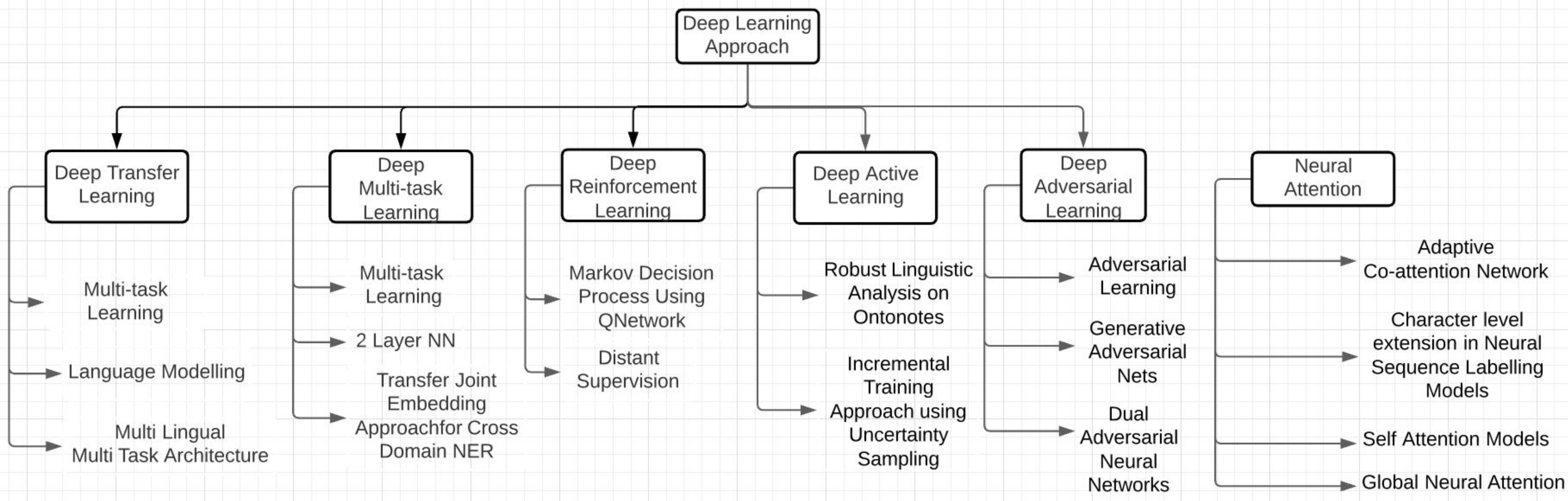
Essentially, iterative approaches that start with patterns, dictionaries or unambiguous entities start with high precision but low recall. Recall is then improved with already tagged entities and feedback. Manually annotated data is typically reserved for model evaluation. [3]



Approaches to NER



Approaches to NER
Rule Based Approach
Feature Based Supervised Learning Approach
Unsupervised Learning Approach
Deep Learning Approach
LaSIE II
NetOwl
Facile
SAR
FASTUS
LTG
Clustering
WordNet Type Signature
KNOWITALL
HMM
Decision Trees
MEM
SVM
CRF



Deep Learning Approach
Deep Transfer Learning
Deep Multi-task Learning
Deep Reinforcement Learning
Deep Active Learning
Deep Adversarial Learning
Neural Attention
Multi-task Learning
Language Modelling
Multi Lingual Multi Task Architecture
Multi-task Learning
2 Layer NN
Transfer Joint Embedding Approachfor Cross Domain NER
Markov Decision Process Using QNetwork
Distant Supervision
Robust Linguistic Analysis on Ontonotes
Incremental Training Approach using Uncertainty Sampling
Adversarial Learning
Generative Adversarial Nets
Dual Adversarial Neural Networks
Adaptive Co-attention Network
Character level extension in Neural Sequence Labelling Models
Self Attention Models
Global Neural Attention

Approaches to NER (1-2 para per approach)

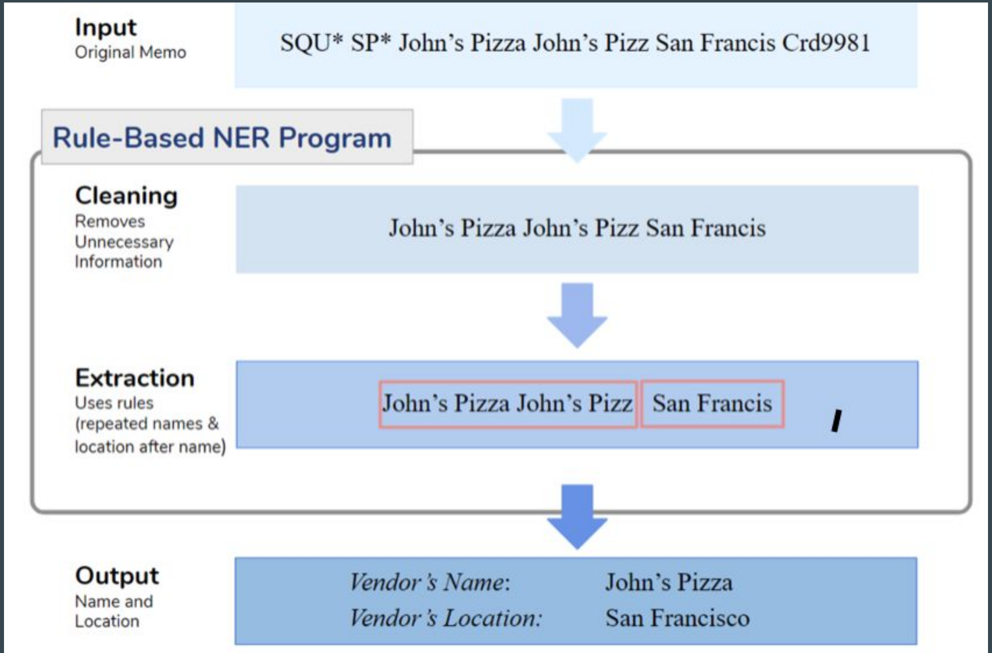
- Rule-based
  - LaSIE-II [1], 1998
  - NetOwl [2], 1996
  - Facile [3], 1998
  - SAR [4], 1998
  - FASTUS [5], 1995
  - LTG [6], 1999
- Unsupervised-based
  - Clustering [7], 2007
  - WordNet type signature [8], 2002
  - KNOWITALL [9], 2005
- Features-based
  - HMM [10], 1996
  - Decision Trees [11], 1986
  - MEM [12], 1989
  - SVM [13], 1998
  - CRF [14], 2001

Approaches to NER

- Deep Learning-based
  - Deep Multi-task Learning
    - Multitask Learning[1], 1997
    - Language Modelling[2], 2017
    - Multi-lingual Multi-task Architecture[3] , 2018
  - Deep Transfer Learning
    - Transfer Joint Embedding Approach for Cross-Domain NER [4], 2013
    - 2-layer Neural Network (Corr b/w Src and Target) [5], 2016
    - Multi-task Learning [6], 2017
  - Deep Reinforcement Learning
    - Markov Decision Process using deep Q-network [7], 2016
    - Distant Supervision [8], 2018
  - Deep Active Learning
    - Incremental Training Approach using Uncertainty Sampling [9, 10], 2017
    - Robust Linguistic Analysis on Ontonotes [11], 2013
  - Deep Adversarial Learning
    - Adversarial Learning [12], 2005
    - Generative Adversarial Nets [13], 2014
    - Dual Adversarial Neural Networks [14], 2018
  - Neural Attention
    - Adaptive Co-attention Network [15], 2018
    - Character level extension in Neural Sequence Labelling Models [16], 2016
    - Self Attention Models [17], 2017
    - Global Neural Attention [18], 2018

Rule Based Approach

- Rely on hand-crafted rules
- Rules can be based on domain-specific gazetteers and syntactic-lexical patterns
- The program generally consists of two parts – cleaning and extraction.
- In the cleaning process, we remove irrelevant information from each memo to ease the extraction of names and locations.
- In the extraction process, we identify the information from the "cleaned" memo.
- For both cleaning and extraction processes, we recognize the irrelevant information and identify the patterns of occurrences of names and locations for extraction



Some works in Rule Based Approaches

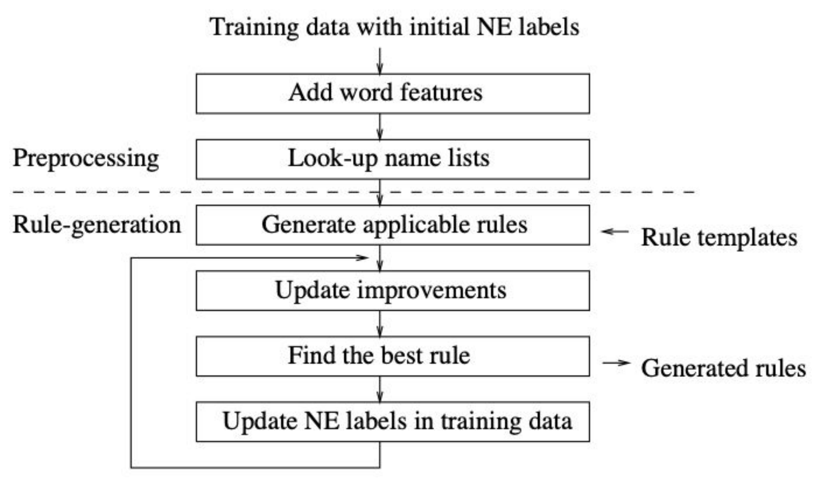
Kim [1] proposed to use Brill rule inference approach for speech input. This system generates rules automatically based on Brill's part-of-speech tagger.

In biomedical domain, Hanisch et al. [2] proposed ProMiner, which leverages a pre-processed synonym dictionary to identify protein mentions and potential gene in biomedical text.

Quimbaya et al. [3] proposed a dictionary-based approach for NER in electronic health records.

Some other well-known rule-based NER systems include LaSIE-II , NetOwl, Facile, SAR, FASTUS, and LTG systems.

Experimental results show the approach improves recall while having limited impact on precision.



Advantages:

- Declarative Approach (Highly Transparent)
- Readability and Maintainability
- Indirectly incorporate the domain knowledge
- Better to complement with ML approaches (Create Training Data)
- Directly translates the manual labor to quality of rules

Disadvantages:

- A lot of manual work
- Time Consuming
- Less learning Capacity
- Complex Domains - Complex pattern Identification
- Rule-based systems work very well when lexicon is exhaustive. Due to domain-specific rules and incomplete dictionaries, high precision and low recall are often observed from such systems, and the systems cannot be transferred to other domains.
- Difficult to incorporate minor deviations in the rules

Feature-based Supervised Learning Approaches

- Applying supervised learning, NER is cast to a multi-class classification or sequence labeling task.
- Given annotated data samples, features are carefully designed to represent each training example.
- Machine learning algorithms are then utilized to learn a model to recognize similar patterns from unseen data.
- Feature engineering is critical in supervised NER systems. Feature vector representation is an abstraction over text where a word is represented by one or many Boolean, numeric, or nominal values
- Word-level features list lookup features and document and corpus features have been widely used in various supervised NER systems.
- Based on these features, many machine learning algorithms have been applied in supervised NER, including Hidden Markov Models (HMM) , Decision Trees, Maximum Entropy Models, Support Vector Machines (SVM), and Conditional Random Fields (CRF).

Some works

- Bikel et al. [4] proposed the first HMM-based NER system, named IdentiFinder, to identify and classify names, dates, time expressions, and numerical quantities.
- Szarvas et al. [5] developed a multilingual NER system by using C4.5 decision tree and AdaBoostM1 learning algorithm
- Borthwick et al. [6] proposed "maximum entropy named entity" (MENE) by applying the maximum entropy theory. MENE is able to make use of an extraordinarily diverse range of knowledge sources in making its tagging decisions.
- McNamee and Mayfield [7] used 1000 language-related and 258 orthography and punctuation features to train SVM classifiers. Each classifier makes binary decision whether the current token belongs to one of the eight classes,

Unsupervised Learning Approaches

- Unsupervised learning refers to the use of artificial intelligence algorithms to identify patterns in data sets containing data points that are neither classified nor labeled.
- A typical approach of unsupervised learning is clustering that can be used in Unsupervised NER.
- Clustering-based NER systems extract named entities from the clustered groups based on context similarity. The key idea is that lexical resources, lexical patterns, and statistics computed on a large corpus can be used to infer mentions of named entities
- Collins et al. [8] observed that use of unlabeled data reduces the requirements for supervision to just 7 simple "seed" rules. The authors then presented two unsupervised algos for named entity classification.
- Nadeau et al. [9] proposed an unsupervised system for gazetteer building and named entity ambiguity resolution. This system combines entity extraction and disambiguation based on simple yet highly effective heuristics.
- Zhang and Elhadad [10] proposed an unsupervised approach to extracting named entities from biomedical text. Instead of supervision, their model resorts to terminologies, corpus statistics and shallow syntactic knowledge. Experiments on two mainstream biomedical datasets demonstrate the effectiveness and generalizability of their unsupervised approach.

Ex: BERT

NER is done unsupervised without labeled sentences using a BERT model that has only been trained unsupervised on a corpus with the masked language model objective.

The model has an F1-score of
97% on a small data set of 25 entity types *(wiki-text corpus)*
86% for person and location on CoNLL-2003 corpus.
It has a lower F1-score of 76% for person, location and organization on CoNLL-2003 corpus largely because of the entity ambiguity inherent in the sentences

Both these tests were done without any pre-training/fine-tuning of model on the data it was tested on.

BERT's mask language model head can predict candidate words for the masked positions, given its training objective described i.e, it learns by predicting words that have been blanked out in a sentence. This learning is then used during inference, to output a prediction for a masked term in a sentence, where the prediction is a probability distribution over BERT's fixed vocabulary of words.

Deep Learning Techniques

In recent years, DL-based NER models become dominant and achieve state-of-the-art results. Compared to feature- based approaches, deep learning is beneficial in discovering hidden features automatically.

There are three core strengths of applying deep learning techniques to NER.

First, NER benefits from the non-linear transformation, which generates non-linear mappings from input to output. Compared with linear models (e.g., log- linear HMM and linear chain CRF), DL-based models are able to learn complex and intricate features from data via non-linear activation functions.

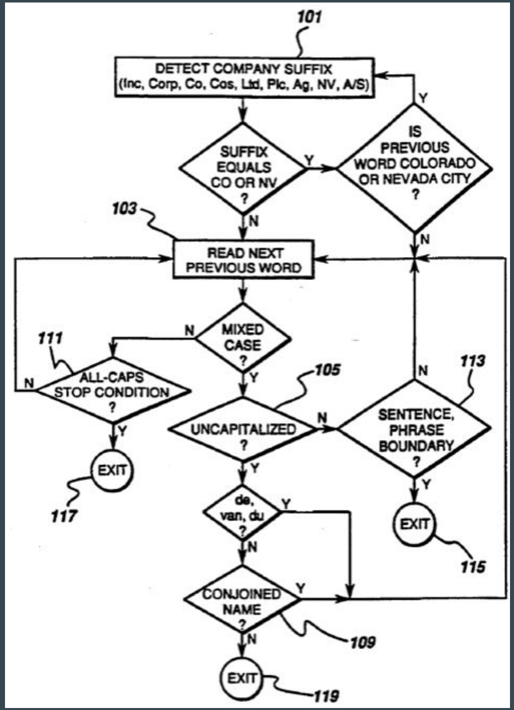
Second, deep learning saves significant effort on designing NER features. The traditional feature-based approaches require considerable amount of engineering skill and domain expertise. DL-based models, on the other hand, are effective in automatically learning useful representations and underlying factors from raw data.

Third, deep neural NER models can be trained in an end-to-end paradigm, by gradient descent. This property enables us to design possibly complex NER systems.

Evolution of NER Techniques

Some Milestones

1991	Lisa Rau [11] implements an algorithm to extract company names from financial news. It's a combination of heuristics, exception lists and extensive corpus analysis. In subsequent retrieval tasks, the algorithm also looks at most likely variations of names
1996	The term **Named Entity** is first used at the 6th Message Understanding Conference (MUC).
2003	Hammerton [12] applies **Long Short-Term Memory (LSTM)** neural network to NER. He finds significantly better performance for German but a disappointing baseline performance for English. Words and their sequences are represented using SARDNET. The algorithm operates in two passes. In the first pass, information is gathered. In the second pass, the algorithm disambiguates and outputs the named entities.
2009	Ratinov and Roth [13] address some design challenges for NER. They note that NER is knowledge intensive in nature. Therefore, they use 30 **gazetteers**, 16 of which are extracted from Wikipedia. In general, these are high-precision, low-recall lists. They are shown to be effective for web pages, where there's less contextual information. Expressive features and gazetteers enable unsupervised learning.

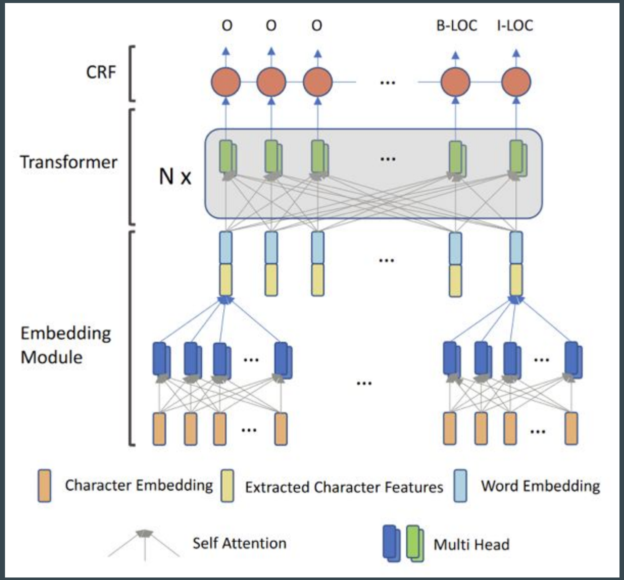


Some Milestones

2011	Tweets have the problem of insufficient information. NER models trained on news articles also do poorly on tweets. This can be solved by domain adaptation or semi-supervised learning from lots of unlabelled data. Liu et al. [14] approach this problem with a combination of **K-Nearest Neighbour (KNN) classifier** and **Conditional Random Field (CRF) labeller**. KNN captures global coarse evidence while CRF captures fine-grained information from a single tweet. The classifier is retrained based on recently labelled tweets. Gazetteers are also used.
2015	Santos and Guimarães [15] extend the work of Collobert et al. by considering **character-level representations** using a convolutional layer. They note that "word-level embeddings capture syntactic and semantic information, character-level embeddings capture morphological and shape information". The use of both gives best results.
2016	Ma and Hovy [16] achieve state-of-the-art F1 score of 91.21 for NER on CoNLL 2003 dataset. Their approach requires no feature engineering or specific data pre-processing. They use CNN to obtain character-level representations to capture morphological features such as prefixes and suffixes. Character embeddings are randomly initialized and then used to obtain character-level representations. Chiu and Nichols present a similar work that also uses word-level features.
2018	Güngör et al. [17] study NER for morphologically rich languages such as Turkish, Finnish, Czech and Spanish. Word morphology is important for these languages. This is in contrast to English that gets useful information from syntax and word n-grams. They therefore propose a model that uses **morphological embedding**. This is combined with character-based and word embeddings. Both character-based and morphological embeddings are derived using separate BiLSTMs.

Some Milestones

2019	Francis et al. [18] make use of BERT language model for **transfer learning** in NER. For fine tuning, either softmax or CRF layer is used. They find BERT representations perform best in combination with character-level representation and word embeddings. Without BERT, they also show competitive performance when character-level representation and word embeddings are gated via an attention layer.
2019	Yan et al. [19] note that the traditional transformer architecture is not quite as good for NER as it is for other NLP tasks. They customize transformer architecture and achieve state-of-the-art results, beating prevailing BiLSTM models. They call it **Transformer Encoder for NER (TENER)**. While traditional transformer uses position embedding, directionality is lost. TENER uses relative position encoding to capture distance and direction. Smoothing and scaling of traditional transformer is seen to attend to noisy information. TENER therefore uses sharp unscaled attention.



Evaluation of NER

Precision is the fraction of relevant instances among retrieved instances.

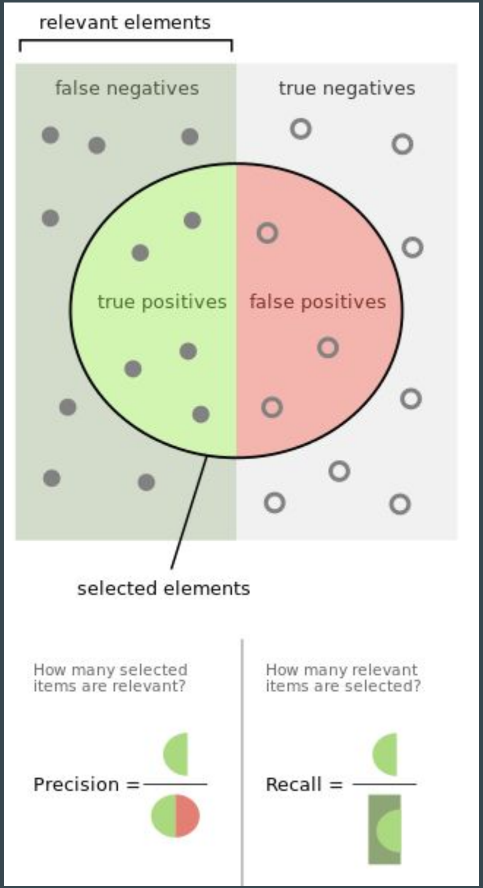
P = tp/(tp+fp)

Recall is the fraction of the relevant instances that were retrieved.

R = tp/(tp+fn)

F1 is a measure that combines precision and recall as the harmonic mean that weighs precision and recall equally,

F1 = 2*P*R/ (P+R)



Dataset Tagging - IOB format

The **IOB format** (short for inside, outside, beginning) is a common tagging format for tagging tokens in a chunking task in computational linguistics (ex. named-entity recognition).

The I- prefix before a tag indicates that the tag is inside a chunk. An O tag indicates that a token belongs to no chunk. The B- prefix before a tag indicates that the tag is the beginning of a chunk that immediately follows another chunk without O tags between them.

Other Tagging Schemes include BIOES/BILOU, where 'E' and 'L' denotes Last or Ending character is such a sequence and 'S' denotes Single element or 'U' Unit elements.

Ex:

IOB Tagging

```
Alex I-PER
is O
going O
to O
Los I-LOC
Angeles I-LOC
in O
California I-LOC
```

IOB2 Tagging

```
Alex B-PER
is O
going O
to O
Los B-LOC
Angeles I-LOC
in O
California B-LOC
```

BIOES Tagging

```
Alex S-PER
is O
going O
with O
Marty B-PER
A. I-PER
Rick E-PER
to O
Los B-LOC
Angeles E-LOC
```

State-of-the-art NER Systems

Datasets Available for NER

- A total of 69 data sets are available for NER (Best Matched NER Task) There are multiple datasets that are available for other tasks such as QnA, Machine Translation, Entity Linking etc.
- 48 of the data sets are of text modality, 7 of medical and 1 of each Audio(MEDIA french corpus), Graphs, Images and Speech.
- 27 of the data sets are in English Language, 7 in Chinese, 5 in German, 3 in Danish, 2 or 1 in most of the other languages
- Some of the Examples of the data sets are: CoNLL-2003, BC5CDR (Biocreative cs CDR Corpus), GENIA(Biomedical Literature), CoNLL 2002, OntoNotes 5.0, SciERC(Scientific), ACE 2005(Multilingual) etc

CoNLL-2003

- Introduced by Sang et al. [20] in Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition
- CoNLL-2003 is a named entity recognition dataset released as a part of CoNLL-2003 shared task: language-independent named entity recognition.
- The data consists of eight files covering two languages: English and German. For each of the languages there is a training file, a development file, a test file and a large file with unannotated data.

English data	Articles	Sentences	Tokens	LOC	MISC	ORG	PER
Training set	946	14,987	203,621	7140	3438	6321	6600
Development set	216	3,466	51,362	1837	922	1341	1842
Test set	231	3,684	46,435	1668	702	1661	1617

German data	Articles	Sentences	Tokens	LOC	MISC	ORG	PER
Training set	553	12,705	206,931	4363	2288	2427	2773
Development set	201	3,068	51,444	1181	1010	1241	1401
Test set	155	3,160	51,943	1035	670	773	1195

- Each word has been put on a separate line and there is an empty line after each sentence.
- The first item on each line is a word, the second a part-of-speech (POS) tag, the third a syntactic chunk tag and the fourth the named entity tag.
- The named entity tags have the format I-TYPE which means that the word is inside a phrase of type TYPE.
- Only if two phrases of the same type immediately follow each other, the first word of the second phrase will have tag B-TYPE to show that it starts a new phrase.
- A word with tag O is not part of a phrase.
- train.txt, valid.txt and test.txt in the data folder have sentences along with their tags. We need only the named entity tags. We extract the words along with their named entities into an array.
- This tagging scheme is the IOB scheme.

U.N.	NNP	I-NP	I-ORG
official	NN	I-NP	O
Ekeus	NNP	I-NP	I-PER
heads	VBZ	I-VP	O
for	IN	I-PP	O
Baghdad	NNP	I-NP	I-LOC
.	.	O	O

Named Entity Recognition on CoNLL 2003 (English)

Model	Paper	F1	Year
ACE+document-context	Automated Concatenation of Embeddings for Structured Prediction	94.6	2021
LUKE	LUKE: Deep Contextualized Entity Representations with Entity-aware Self-attention	94.3	2020
FLERT XLM-R	FLERT: Document-Level Features for Named Entity Recognition	94.09	2020
PL-Marker	Pack Together: Entity and Relation Extraction with Levitated Marker	94.0	2021
CL-KL	Improving Named Entity Recognition by External Context Retrieving and Cooperative Learning	93.85	2021

Ontonotes 5.0

- OntoNotes Release 5.0 is the final release of the OntoNotes project, a collaborative effort between BBN Technologies, the University of Colorado, the University of Pennsylvania and the University of Southern California's Information Sciences Institute.
- The goal of the project was to annotate a large corpus comprising various genres of text (news, conversational telephone speech, weblogs, usenet newsgroups, broadcast, talk shows) in three languages (English, Chinese, and Arabic) with structural information (syntax and predicate argument structure) and shallow semantics (word sense linked to an ontology and coreference).
- This publication consists of 2.9 million words with counts shown in the table..
- The Named entity classes are PERSON, NORP, FACILITY, ORG, GPE, LOCATION, PRODUCT, EVENT, WORK OF ART, LAW, LANGUAGE etc (18 entities)
- This annotation method uses markup tags using angle brackets for defining named entity
- *<ENAMEX TYPE="ORG">Disney</ENAMEX> is a global brand .*

```
7    over
8    the
        coref: IDENT        000-75 8-9    the weekend
        name:  DATE                8-9    the weekend
9    weekend
10   what
11   his
        coref: IDENT        000-69 11-11  his
12   U.S.
        name:  GPE                 12-12  U.S.
13   antagonists
14   have
15   failed
        sense: fail-v.1
        prop:  fail.01
         v         * -> 15:0 failed
         ARG2      * -> 19:0 *T*-2
                   * -> 10:1 what
         ARG1      * -> 11:1 his U.S. antagonists
         LINK-SLC  * -> 10:1 what
                   * -> 21:2 *PRO* revive a constituency for the Contra rebels
16   *-1
17   to
```

	Arabic	English	Chinese
News	300k	625k	250k
BN	n/a	200k	250k
BC	n/a	200k	150k
Web	n/a	300k	150k
Tele	n/a	120k	100k
Pivot	n/a	n/a	300

Named Entity Recognition on Ontonotes v5 (English)

Model	Paper	F1	Year
BERT-MRC+DSC	Dice Loss for Data-imbalanced NLP Tasks (Needs Extra Training Data)	92.07	2019
PL-Marker	Pack Together: Entity and Relation Extraction with Levitated Marker	91.9	2021
Biaffine-NER	Named Entity Recognition as Dependency Parsing	91.3	2020
BERT-MRC	A Unified MRC Framework for Named Entity Recognition (Needs Extra Training Data)	91.11	2019
Syn-LSTM + BERT	Better Feature Integration for Named Entity Recognition	90.85	2021

WNUT 2017 (WNUT 2017 Emerging and Rare entity recognition)

- Introduced by Derczynski et al. [21] in Results of the WNUT2017 Shared Task on Novel and Emerging Entity Recognition
- This dataset focuses on identifying unusual, previously-unseen entities in the context of emerging discussions. Named entities form the basis of many modern approaches to other tasks (like event clustering and summarisation), but recall on them is a real problem in noisy data (Derczynski et al 2015) - even among annotators.
- This drop tends to be due to novel entities and surface forms. ((Augenstein et al 2017)).
- This task will evaluate the ability to detect and classify novel, emerging, singleton named entities in noisy text.
- The goal of this task is to provide a definition of emerging and of rare entities, and based on that, also datasets for detecting these entities. The WNUT 2017 shared task poses this challenge directly to participants, with turbulent data containing few repeated entities, drawn from rapidly-changing text types or sources of non-mainstream entities.

ACE 2005

- ACE 2005 Multilingual Training Corpus was developed by the Linguistic Data Consortium (LDC) and contains approximately 1,800 files of mixed genre text in English, Arabic, and Chinese annotated for entities, relations, and events.
- This represents the complete set of training data in those languages for the 2005 Automatic Content Extraction (ACE) technology evaluation.
- The genres include newswire, broadcast news, broadcast conversation, weblog, discussion forums, and conversational telephone speech.
- The objective of the ACE program was to develop automatic content extraction technology to support automatic processing of human language in text form.

BC5CDR

- BC5CDR corpus consists of 1500 PubMed articles with 4409 annotated chemicals, 5818 diseases and 3116 chemical-disease interactions.
- BC5CDR is a popular biomedical NER dataset.

Named Entity Recognition on WNUT 2017

Model	Paper	F1	Year
CL-KL	Improving Named Entity Recognition by External Context Retrieving and Cooperative Learning	60.45	2021
TNER -xlm-r-large	T-NER: An All-Round Python Library for Transformer-based Named Entity Recognition	58.5	2021
BERTweet	BERTweet: A pre-trained language model for English Tweets	56.5	2020
SpanNer	SpanNER: Named Entity Re-/Recognition as Span Prediction	52.97	2021
Truecase	Robust Named Entity Recognition with Truecasing Pretraining	52.3	2019

Named Entity Recognition on ACE 2005

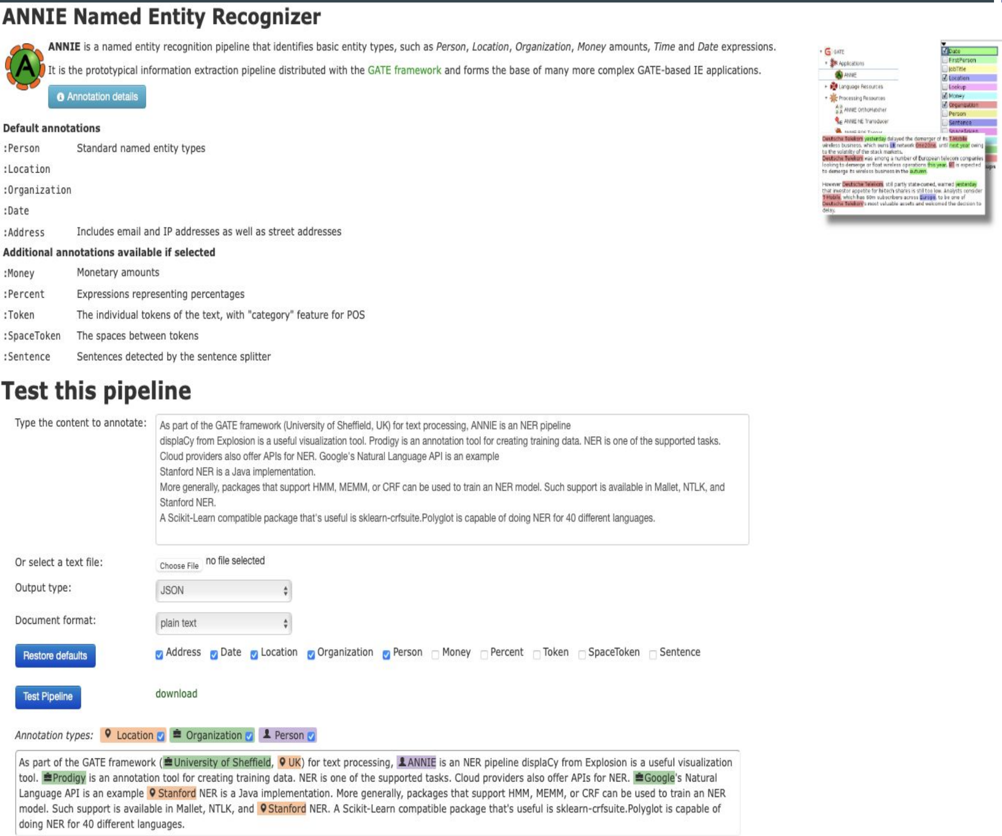
Model	Paper	F1	Year
Ours: cross-sentence ALB	A Frustratingly Easy Approach for Entity and Relation Extraction	90.9	2020
BERT-MRC	A Unified MRC Framework for Named Entity Recognition (Needs Extra Training Data)	86.88	2019
Locate and Label	Locate and Label: A Two-stage Identifier for Nested Named Entity Recognition	86.67	2021
BoningKnife	BoningKnife: Joint Entity Mention Detection and Typing for Nested NER via prior Boundary Knowledge	85.46	2021
Biaffine-NER	Named Entity Recognition as Dependency Parsing	85.4	2020

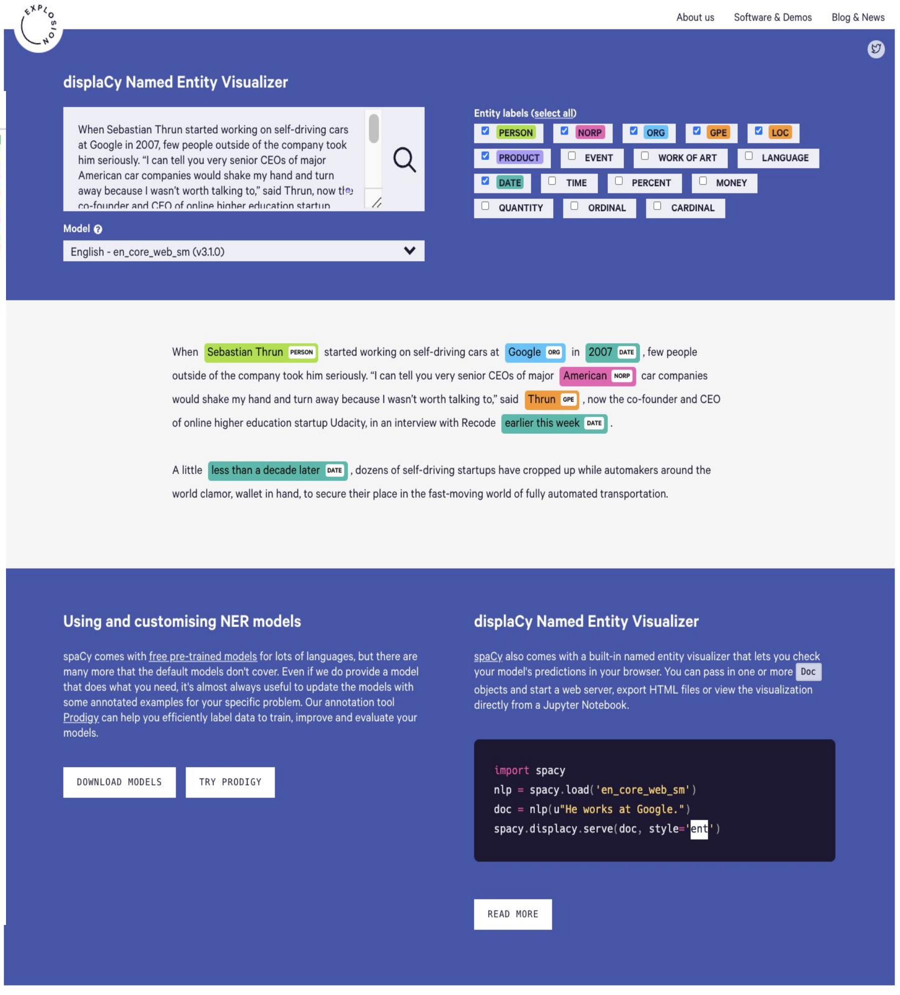
Named Entity Recognition on BC5CDR

Model	Paper	F1	Year
CL-L2	Improving Named Entity Recognition by External Context Retrieving and Cooperative Learning	90.9	2021
ELECTRAMed	ELECTRAMed: a new pre-trained language representation model for biomedical NLP (Used Extra Training Data)	86.88	2021
NER+PA+RL	Reinforcement-based denoising of distantly supervised NER with partial annotation (Used Extra Training Data)	86.67	2019
BLSTM-CNN-Char	Biomedical Named Entity Recognition at Scale	85.46	2020
Spark NLP	Biomedical Named Entity Recognition at Scale	85.4	2020

Resources available for NER Research

- As part of the GATE framework (University of Sheffield, UK) for text processing, ANNIE is an NER pipeline
- displaCy from Explosion is a useful visualization tool. Prodigy is an annotation tool for creating training data. NER is one of the supported tasks.
- Cloud providers also offer APIs for NER. Google's Natural Language API is an example
- Stanford NER is a Java implementation.
- More generally, packages that support HMM, MEMM, or CRF can be used to train an NER model. Such support is available in Mallet, NLTK, and Stanford NER.
- A Scikit-Learn compatible package that's useful is sklearn-crfsuite.Polyglot is capable of doing NER for 40 different languages.


ANNIE Named Entity Recognizer
ANNIE is a named entity recognition pipeline that identifies basic entity types, such as Person, Location, Organization, Money amounts, Time and Date expressions.
It is the prototypical information extraction pipeline distributed with the GATE framework and forms the base of many more complex GATE-based IE applications.
Annotation details
Default annotations
:Person Standard named entity types
:Location
:Organization
:Date
:Address Includes email and IP addresses as well as street addresses
Additional annotations available if selected
:Money Monetary amounts
:Percent Expressions representing percentages
:Token The individual tokens of the text, with "category" feature for POS
:SpaceToken The spaces between tokens
:Sentence Sentences detected by the sentence splitter
Test this pipeline
Type the content to annotate:
As part of the GATE framework (University of Sheffield, UK) for text processing, ANNIE is an NER pipeline
displaCy from Explosion is a useful visualization tool. Prodigy is an annotation tool for creating training data. NER is one of the supported tasks.
Cloud providers also offer APIs for NER. Google's Natural Language API is an example
Stanford NER is a Java implementation.
More generally, packages that support HMM, MEMM, or CRF can be used to train an NER model. Such support is available in Mallet, NTLK, and Stanford NER.
A Scikit-Learn compatible package that's useful is sklearn-crfsuite.Polyglot is capable of doing NER for 40 different languages.
Or select a text file: Choose File no file selected
Output type: JSON
Document format: plain text
Restore defaults
Address Date Location Organization Person Money Percent Token SpaceToken Sentence
Test Pipeline
download
Annotation types: Location Organization Person
As part of the GATE framework (University of Sheffield, UK) for text processing, ANNIE is an NER pipeline displaCy from Explosion is a useful visualization tool. Prodigy is an annotation tool for creating training data. NER is one of the supported tasks. Cloud providers also offer APIs for NER. Google's Natural Language API is an example Stanford NER is a Java implementation. More generally, packages that support HMM, MEMM, or CRF can be used to train an NER model. Such support is available in Mallet, NTLK, and Stanford NER. A Scikit-Learn compatible package that's useful is sklearn-crfsuite.Polyglot is capable of doing NER for 40 different languages.


About us
Software & Demos
Blog & News
displaCy Named Entity Visualizer
When Sebastian Thrun started working on self-driving cars at Google in 2007, few people outside of the company took him seriously. "I can tell you very senior CEOs of major American car companies would shake my hand and turn away because I wasn't worth talking to," said Thrun, now the co-founder and CEO of online higher education startup
Entity labels (select all)
PERSON NORP ORG GPE LOC
PRODUCT EVENT WORK OF ART LANGUAGE
DATE TIME PERCENT MONEY
QUANTITY ORDINAL CARDINAL
Model
English - en_core_web_sm (v3.1.0)
When Sebastian Thrun PERSON started working on self-driving cars at Google ORG in 2007 DATE, few people outside of the company took him seriously. "I can tell you very senior CEOs of major American NORP car companies would shake my hand and turn away because I wasn't worth talking to," said Thrun GPE, now the co-founder and CEO of online higher education startup Udacity, in an interview with Recode earlier this week DATE.
A little less than a decade later DATE, dozens of self-driving startups have cropped up while automakers around the world clamor, wallet in hand, to secure their place in the fast-moving world of fully automated transportation.
Using and customising NER models
spaCy comes with free pre-trained models for lots of languages, but there are many more that the default models don't cover. Even if we do provide a model that does what you need, it's almost always useful to update the models with some annotated examples for your specific problem. Our annotation tool Prodigy can help you efficiently label data to train, improve and evaluate your models.
DOWNLOAD MODELS
TRY PRODIGY
displaCy Named Entity Visualizer
spaCy also comes with a built-in named entity visualizer that lets you check your model's predictions in your browser. You can pass in one or more Doc objects and start a web server, export HTML files or view the visualization directly from a Jupyter Notebook.
import spacy
nlp = spacy.load('en_core_web_sm')
doc = nlp(u"He works at Google.")
spacy.displacy.serve(doc, style='ent')
READ MORE

Off-the-shelf NER tools offered by academia and industry projects.

NER System	URL
StanfordCoreNLP	https://stanfordnlp.github.io/CoreNLP/
OSU Twitter NLP	https://github.com/aritter/twitter_nlp
Illinois NLP	http://cogcomp.org/page/software/
NeuroNER	http://neuroner.com/
NERsuite	http://nersuite.nlplab.org/
Polyglot	https://polyglot.readthedocs.io
Gimli	http://bioinformatics.ua.pt/gimli
spaCy	https://spacy.io/api/entityrecognizer
NLTK	https://www.nltk.org
OpenNLP	https://opennlp.apache.org/
LingPipe	http://alias-i.com/lingpipe-3.9.3/
AllenNLP	https://demo.allennlp.org/
IBM Watson	https://natural-language-understanding-demo.ng.bluemix.net
FG-NER	https://fgner.alt.ai/extractor/
Intellexer	http://demo.intellexer.com/
Repustate	https://repustate.com/named-entity-recognition-api-demo
AYLIEN	https://developer.aylien.com/text-api-demo
Dandelion API	https://dandelion.eu/semantic-text/entity-extraction-demo
displaCy	https://explosion.ai/demos/displacy-ent
ParallelDots	https://www.paralleldots.com/named-entity-recognition
TextRazor	https://www.textrazor.com/named_entity_recognition

Applications of NER

QnA:

Most QA systems gradually reduce the amount of data they need to consider in several phases.

Ex: When the system receives a user question, it first selects a set of relevant documents, and then filters out irrelevant pieces of text of these documents gradually until the answer is found.

The NER is typically used as an aid to filter out strings that do not contain the answer. The NER is used to single out the entity types appearing in a text fragment.
If a piece of text does not have any entity with a type compatible with the type of the expected answer, the text is discarded or heavily penalised.
The desiderata of a NER are related with the range of entities to detect and with the recall of the system.

Content Recommendation Systems:

NER can be used in developing algorithms for recommender systems that make suggestions based on our search history or on our present activity. This is achieved by extracting the entities associated with the content in our history or previous activity and comparing them with the label assigned to other unseen content. Thus we frequently see the content of our interest.

Optimizing Search Engine Algorithms:

When designing a search engine algorithm, It would be an inefficient and computational task to search for an entire query across the millions of articles and websites online, an alternate way is to run a NER model on the articles once and store the entities associated with them permanently. Thus for a quick and efficient search, the key tags in the search query can be compared with the tags associated with the website articles

Automatically Summarizing Resumes:

The majority of resume summarizing tools use the NER software which helps it to retrieve such information. A lot of resumes are excessively populated in detail, of which, most of the information is irrelevant to the evaluator.

Using the NER model, the relevant information to the evaluator can be easily retrieved from them thereby simplifying the effort required in shortlisting candidates among a pile of resumes.

Classifying content for news providers

A large amount of online content is generated by the news and publishing houses on a daily basis and managing them correctly can be a challenging task.

NER can automatically scan entire articles and help in identifying and retrieving entities discussed in them. Thus articles are automatically categorized in defined hierarchies and the content is easily discovered.

Simplifying Customer Support:

Using NER we can recognize relevant entities in customer complaints and feedback such as Product specifications, department, or company branch location so that the feedback is classified accordingly and forwarded to the appropriate department responsible for the identified product.

Booking.com - Customer Communications

Compared 3 models

1. Structural SVM
   Used MITIE, an open source NLP library from MIT
2. Recurrent Neural Networks with word embeddings
   Used the paper "Using Recurrent Neural Networks for Slot filling in Spoken Language Understanding" [22] as an inspiration
3. Learning2Search
   This method comes from the team behind Vowpal Wabbit[23] , an open-source online ML system library. The L2S approach has been demoed as a sequential decision-making approach to NER.

- The team put together a small use case that allowed them to compare the performance of the three methods:
- *The top 10% of clicked destinations were used to build a sample dataset for the prototype models presented.* They identified three entities for labeling.
  1. destination (dest)
  2. facility (fac)
  3. property type (prop_type)
- *The whole dataset created using the different combinations of destinations, facilities and property types was around 200,000 rows and 20% of it was used as a test set to evaluate the models.*
- Ultimately, they concluded that L2S was by far the best model for their purposes, scoring better overall on precision, recall, and f1 score. L2S was followed by structural SVM.

class	precision	recall	f1
Learning2Search			
dest	1.00	0.95	0.97
fac	0.93	1.00	0.96
prop_type	1.00	0.94	0.97
avg / total	0.97	0.97	0.97
Elman RNN			
dest	1.00	1.00	1.00
fac	0.64	1.00	0.78
prop_type	1.00	0.46	0.63
avg / total	0.88	0.81	0.80
MITIE			
dest	1.00	0.76	0.86
fac	0.80	1.00	0.89
prop_type	1.00	1.00	1.00
avg / total	0.93	0.91	0.91

Comparison of MITIE, Elman and L2S

Other Examples

- Zalando: used NER to identify fashion brands, occasions, seasons, colors and clothing items.
- The European Holocaust Research Infrastructure project (EHRI) used NER to automatically parse thousands of transcribed oral testimonials and recognize entities such as names, places, organizations, and events and make a museum catalog more accessible and interactive.
- Machine Translation
- Discover Texts Subject
- Discover Relationship among Entities
- Document Indexing
- Product Reviews
- Entities in Emails (Classification / Information Extraction)

Challenges

- Ambiguities make NER a challenging task. Ex , 'JFK' can refer to former US President John F. Kennedy or his son. **Coreference resolution** is an NLP task that resolves this ambiguity. More common to NER is when the same name refers to different types. For example, 'JFK' can refer to the airport in New York or Oliver Stone's 1991 movie.
- Since entities can span multiple words/tokens, NER needs to identify start and end of multi-token entities. When a word is a qualifier, it may be wrongly tagged. For example, in *Clinton government*, an NER system maytag Clinton as PER without recognizing the noun phrase.
- Same entity can appear in different forms. Differences could be typographical , morphological , syntactic reduction , or abbreviated .
- Spelling Variations
- Foreign Words

Thank You