I will research and review different tools related to text anonymization. These include NER and PII removal tools such as spaCy, Flair, in terms of features, availability, cost, and effectiveness

Name-Entity Recognition (NER) is a discipline of information extraction that identifies the named entities in text. In other words, NER labels sequences of words in a text which are the names of things, such as person and company names… [1] These named entities are classified into categories: people's names, time, location, organization. Here's an example of NER: in the sentence "George Washington visits Washington", "George Washington" is tagged as Person and "Washington" is tagged as Location. On the other hand, Personally identifiable information (PII) is data that can potentially be used to identify a specific individual. Examples of PII include name, address, email, social security number… This information is widely used to commit identity theft, thus it is important to protect such information from unwanted parties. When processing unstructured text data, sometimes the text body might contain PIIs; therefore, one must remove PIIs prior to looking at or analyzing the text. This process is called text anonymization. During the text anonymization process, NER tools are used to detect PIIs. This paper will discuss two popular open-source tools for NER: spaCy and Flair.

SpaCy is an open-source library for advanced NLP. It is available in Python. One of spaCy's modules is NER, its task is to tag proper nouns and numeric entities. The accuracy of spaCy's NER is shown in Figure 1 [2]. These benchmark tests were run in the OntoNotes 5 dataset. Around 2009, the accuracy was approximately 83.5%. This remained as the best accuracy until 2014, when a model built on neural networks was able to achieve better accuracy. The neural network used for entity recognition is convolutional network with Conditional Random Field (CRF) component on top. The NER model of spaCy, en_core_web_lg 2.x, is similar to Strubell et al. 's system. En_core_web_lg was built on GloVe word embeddings, thus its size is large and it takes a while to download, but it was able to achieve similar accuracy to the Strubell et al. 's system. SpaCy also has another model, en_core_web_sm. However, this model does not use pre-trained word vectors, thus its accuracy is lower compared to the en_core_web_lg model. Figure 2 shows comparison between spaCy's models. The 1.x models are linear models which use more features and less aggressive L1 regularization, while the 2.x models are neural network models. Note that the medium-sized 1.x model, en_core_web_md (1GB), has 81.4% accuracy while the small-sized 2.x model, en_core_web_lg (35MB), has 85.3% accuracy, approximately 4% improvement. Thus, we can see the power of neural networks.

SYSTEM	YEAR	TYPE	ACCURACY
spaCy `en_core_web_lg` v2.0.0a3	2017	neural	85.85
Strubell et al.	2017	neural	**86.81**
Chiu and Nichols	2016	neural	86.19
Durrett and Klein	2014	neural	84.04
Ratinov and Roth	2009	linear	83.45

Figure 1

English

MODEL	SPACY	TYPE	UAS	NER F	POS	WPS	SIZE
`en_core_web_sm` 2.0.0	2.x	neural	91.7	85.3	97.0	10.1k	**35MB**
`en_core_web_md` 2.0.0	2.x	neural	91.7	**85.9**	97.1	10.0k	115MB
`en_core_web_lg` 2.0.0	2.x	neural	**91.9**	**85.9**	**97.2**	10.0k	812MB
`en_core_web_sm` 1.2.0	1.x	linear	86.6	78.5	96.6	**25.7k**	50MB
`en_core_web_md` 1.2.1	1.x	linear	90.6	81.4	96.7	18.8k	1GB

Figure 2

Flair is another state-of-the-art open-source NER Python library. Its framework was built on Pytorch, an open source machine learning library based on the Torch library. Figure 3 shows the F1 scores of Flair's pre-trained NER models [3]. The following features makes Flair a state-of-the-art model compared to other similar libraries:

- Flair library uses state-of-the-art common word embeddings (ELMo, GloVe, BERT…) and allows the user to combine these different word embeddings to embed documents.
- Flair library supports different languages: English, German, Dutch, Polish, French, Dutch, and Danish
- Flair library NER model is assisted by contextual string embeddings. This provides the context when trying to detect entities in a text. In NER, context is very important. For example: in the sentence "Bill Clinton is a former US President", Bill is a person's name and in the sentence "My water bill is too high", bill is just a noun. Flair utilizes

the internal principles of a trained character model to draw conclusion about words' meaning depending on the context.

Flair uses a pre-trained bidirectional character language model (LM). It uses this LM to obtain a context embedding for each word; and it achieves this by extracting the first and last character cell states. Then, this word vector is inputted to a vanila bi-directional long short-term-memory with CRF on top of it. This helps Flair have a state-of-the-art result on downstream NER tasks. Figure 4 shows an example of Flair Character Language Model.

Task	Language	Dataset	Flair	Previous best
Named Entity Recognition	English	Conll-03	**93.18** (F1)	*92.22 (Peters et al., 2018)*
Named Entity Recognition	English	Ontonotes	**89.3** (F1)	*86.28 (Chiu et al., 2016)*
Emerging Entity Detection	English	WNUT-17	**49.49** (F1)	*45.55 (Aguilar et al., 2018)*
Part-of-Speech tagging	English	WSJ	**97.85**	*97.64 (Choi, 2016)*
Chunking	English	Conll-2000	**96.72** (F1)	*96.36 (Peters et al., 2017)*
Named Entity Recognition	German	Conll-03	**88.27** (F1)	*78.76 (Lample et al., 2016)*
Named Entity Recognition	German	Germeval	**84.65** (F1)	*79.08 (Hänig et al, 2014)*
Named Entity Recognition	Dutch	Conll-02	**92.38** (F1)	*81.74 (Lample et al., 2016)*
Named Entity Recognition	Polish	PolEval-2018	**86.6** (F1) (Borchmann et al., 2018)	*85.1 (PolDeepNer)*

Figure 3

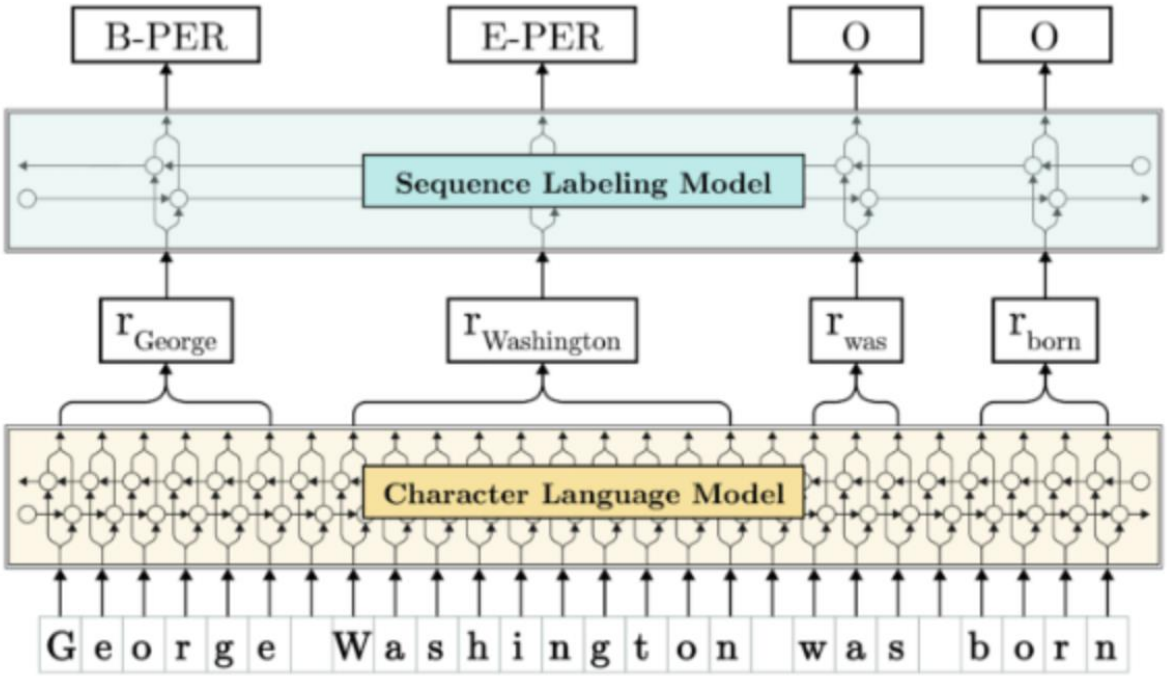

Figure 4

There are other NER tools, both open-source such as Stanford's CoreNLP, OpenNLP, GATE… or commercial software such as Lionbridge, Scale, Appen… Even though these tools feature NER, their applications are different depending on which type of data one would like to analyze. For example, Stanford's CoreNLP works very well on legal dataset [4]. In the above discussion, Flair NER models achieved very high F1 scores on the Conll-03 and Ontonotes datasets; however, it consistently makes incorrect toeknization choices when dealing with trailing punctuation in captured tokens. For example, when analyzing the sentence "We can build on a strong progressive tradition from Franklin Roosevelt to Barack Obama.", it detects the term "Barack Obama." (including the period) as a person. Therefore, the performance for each model depends on the dataset and the use case. Detecting and removing PIIs is only one application of NER. Other applications include getting relevant insights in customer feedback about specific brands or products, improving content recommendations, or classifying topics for news providers.

References:

[1] Stanford Named Entity Recognizer (NER). (2020). The Stanford Natural Language Processing Group. https://nlp.stanford.edu/software/CRF-NER.html

[2] Facts & Figures. (2020). SpaCy. https://spacy.io/usage/facts-figures

[3] F. (2020). flairNLP/flair. GitHub. https://github.com/flairNLP/flair

[4] Which open source NER Model is the best ? Comparing CoreNLP, Spacy and Flair. (2020). Lighttag. https://www.lighttag.io/blog/spacy-vs-stanford/